

# Game Making e Dintorni

Le cose da sapere prima di iniziare a creare un videogioco

# Contenuti

1. Introduzione
2. Cos'è un videogioco
3. Com'è Fatto un Videogioco
4. Cenni sulla Pipeline grafica
5. Conclusioni

# Introduzione

Questa lezione si prefigge lo scopo di fornire un'infarinatura basilare sulla struttura di un videogioco.

Si partirà dalla spiegazione di cos'è un videogioco per poi passare ad illustrarne la sua composizione.

Vedremo anche come si passa da diversi dati (immagini, modelli 3D ecc) ad un'immagine finale.

In ultimo verrà lasciato spazio ad eventuali domande.

# Cos'è un videogioco

Prima di spiegare com'è fatto un videogioco è bene dire cosa si intende per videogioco.

Una definizione potrebbe essere questa:

“Un videogioco è un programma che dati determinati input genera degli output”.

Gli output possono essere di 3 tipi:

- Visivo (l'immagine dello schermo)
- Sonoro (musica, effetti sonori ecc)
- Sensoriali (vibrazione del joypad, force-feedback)

# Cos'è un videogioco

La particolarità di un videogioco è che gli output vengono aggiornati più volte in al secondo (in particolare la grafica).

La frequenza di aggiornamento grafica prende il nome di Frame Per Secondo (o FPS). Più è alto il frame-rate più il videogioco sarà fluido (come in un cartone animato).

Un valore di 25-30 FPS garantisce una fluidità accettabile; l'ideale sarebbe di 50-60 FPS.

# Com'è fatto un videogioco

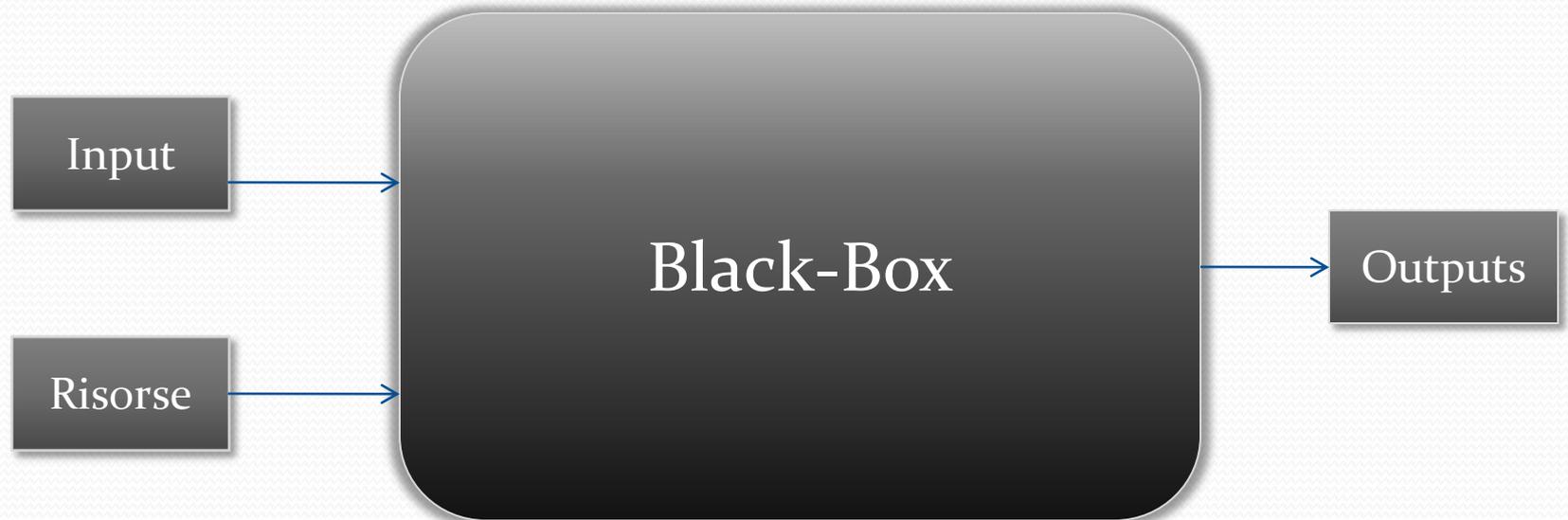
Non c'è una risposta precisa.

La struttura di un videogioco dipende da tanti fattori:

- Genere del videogioco
- Piattaforma di destinazione
- Scelte progettuali
- Ecc...

# Com'è fatto un videogioco

In linea di massima è possibile riassumere la struttura tramite una black-box.



# Com'è fatto un videogioco?

Oppure per avere un riscontro nella vita reale con una macchina.



# Com'è fatto un videogioco?

Il paragone è più che mai azzeccato in quanto in una macchina come in un videogioco si valutano più aspetti:

- Prestazioni
- Consumi
- Estetica
- Comfort/facilità di utilizzo

# Com'è fatto un videogioco?

Inoltre anche il modo di funzionamento è più o meno simile:

- Entrambi richiamano delle risorse quando necessario
- Una macchina con un ottimo motore non sarà per forza un'ottima macchina
- L'utilizzo di una macchina non implica la conoscenza della sua composizione interna (però nel caso la si conosca aiuta)

# Com'è fatto un videogioco?

Per tornare in tema si può dire che un videogioco dipende fortemente dalle risorse (che sono viste come input).

In questo ambito il concetto informatico di “Trash in-Trash out” trova una forte dimostrazione.

Infatti un ottimo motore grafico per dare risultati sbalorditivi necessita di risorse appropriate.

# Trash in-Trash out Esempi

Con i mod è possibile osservare questo fenomeno.



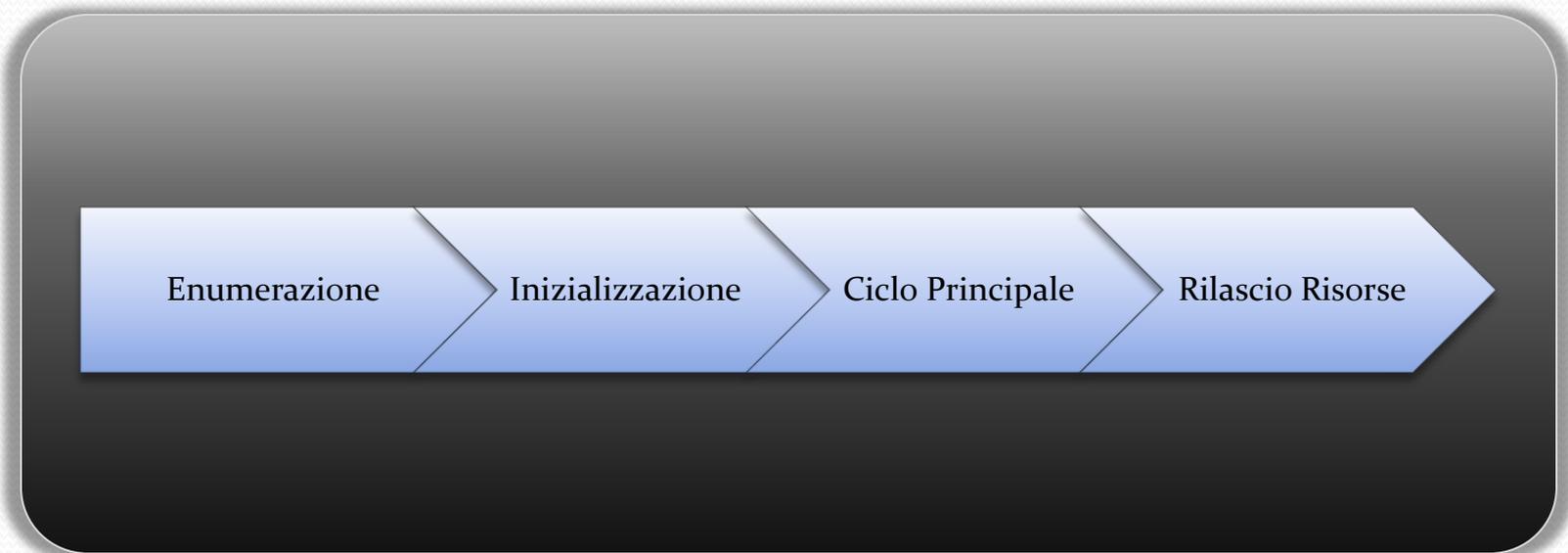
Crysis Very High



Stesso gioco (e impostazioni) ma risorse differenti

# Dentro la “Scatola Nera”

Se apriamo la nostra “scatola” (oppure il cofano della nostra macchina) possiamo trovare una struttura abbastanza comune:



# Dentro la “Scatola Nera”

Le singole fasi vengono eseguite una sola volta.

L'unica eccezione è rappresentata dal Ciclo Principale eseguito fino a quando non terminiamo il gioco.

Le parti indicati sono quelle fondamentali.

Ogni gioco poi, potrà averne delle sue.

# Enumerazione

- E' una fase fondamentale in quanto mi permette di sapere cosa POSSO e cosa NON POSSO fare con l'hardware a disposizione.
- In base ai risultati ottenuti regolerò le impostazioni del gioco
- Riguarda tutto quello che dovrò andare ad utilizzare in seguito (scheda video in primis, joy-pad, scheda audio alle volte)

# Enumerazione

- Nelle console viene eseguita di rado, e in genere solo per input fisici
- Un esempio è rappresentato da Guitar Hero
- Sul computer è presente un' Enumerazione grafica molto intensa.

# Inizializzazione

- Serve per riuscire a dialogare correttamente con l'hardware ed impostare le caratteristiche desiderate (risoluzione ecc)
- In questa fase vengono creati gli oggetti da utilizzare nel Ciclo Main
- L'inizializzazione non riguarda solo l'aspetto grafico

# Ciclo Principale

In questa particolare sezione del programma vengono inserite tutte le istruzioni necessarie per generare i vari input. E' una parte che deve essere molto performante per poter essere eseguita più volte in un secondo.

All'interno troviamo tutte le operazioni necessarie per realizzazione il risultato desiderato.

E' facilmente intuibile che è una parte molto importante.

# Ciclo Principale

Data la sua complessità è utile dividerlo in più blocchi che eseguono diverse operazioni. Ogni blocco poi gestisce in modo automatico le operazioni di sua competenza.

Questa suddivisione è molto vantaggiosa in quanto permette di:

- suddividere il lavoro tra diversi membri del team
- avere un controllo sulle prestazioni
- individuare problemi e correggerli
- esecuzione in parallelo delle singole fasi

# Ciclo Principale

Input

- Fase in cui controllo quali input sono stati forniti (tasti premuti)

Game Play

- In luce degli input forniti aggiorno la logica di gioco (telecamera, fisica, IA ecc)

Grafica

- Una volta aggiornata la scena disegno gli oggetti visibili

# Rilascio delle Risorse

Questa fase viene eseguita nel momento in cui si decide di terminare il videogioco (in poche parole quanto si termina l'esecuzione del ciclo main).

Serve sostanzialmente a liberare la memoria occupata ed a chiudere gli eventuali file aperti.

# Cenni sulla Pipeline Grafica

Parlando di videogiochi è quasi d'obbligo dare qualche cenno sulla Pipeline Grafica.

Essa è quell'insieme di procedimenti che permettono di ottenere un'immagine rappresentativa di uno spazio (2D o 3D) partendo dalla descrizione della sua geometria.

Viene implementata tramite 2 componenti:

- Scheda video: offre l'hardware necessario ad eseguire i vari calcoli matematici necessari (insieme a molto altri)
- API (DirectX, OpenGL ecc): istruiscono la scheda video su come vanno eseguite le operazioni e in che ordine

# Cenni sulla Pipeline Grafica

I calcoli vengono eseguiti su 2 tipi di risorse:

- Vertici: descrivono un modello in uno spazio. Oltre alla posizione di ogni punto del modello possono avere altre informazioni (normali, colore, informazioni sulle texture ecc...)
- Textures: sono delle immagini che vengono utilizzate per rivestire un modello e dare maggiore dettaglio ad esso.

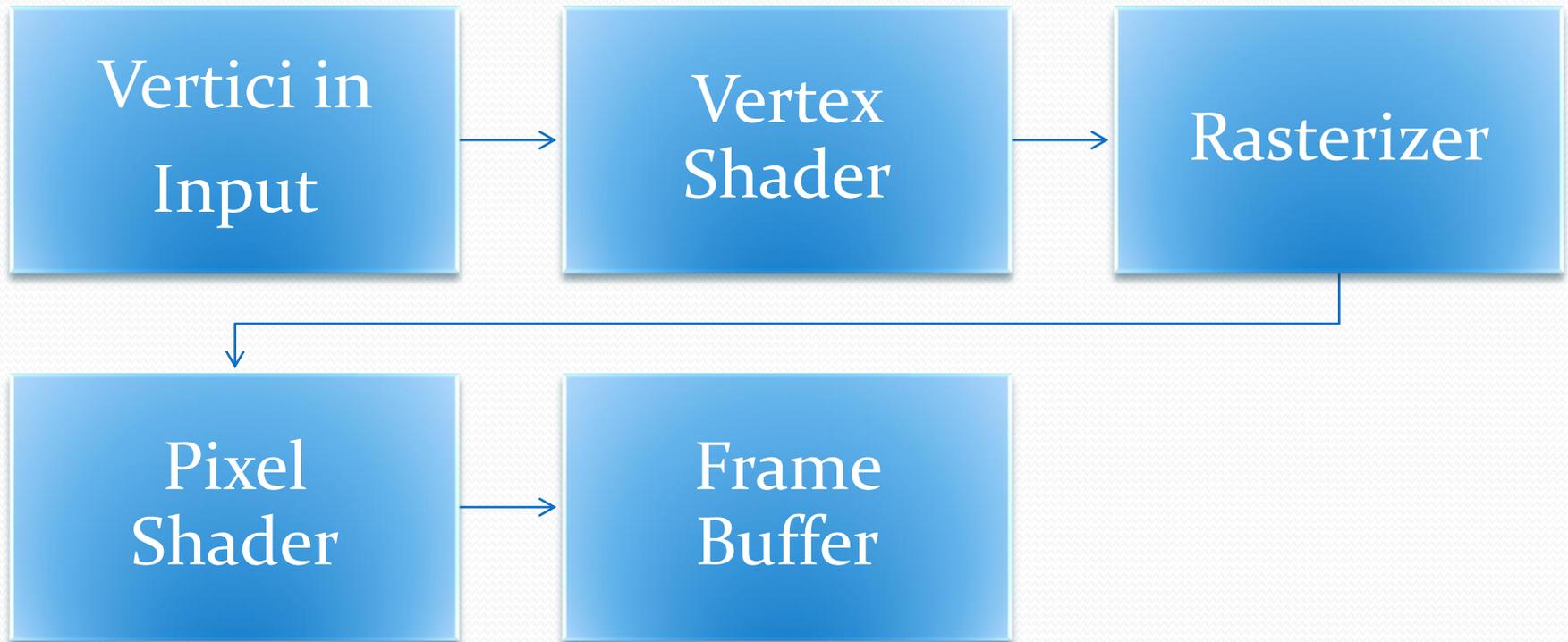
# Cenni sulla Pipeline Grafica

Tutte queste informazioni vengono poi elaborate in diversi passaggi ognuno dei quali può essere programmato per ottenere diversi effetti (luce, ombre ecc).

I passaggi sono fondamentalmente 2:

- Vertex Shader: opera sui vertici e li manipola per poterli rappresentare nello spazio World
- Pixel Shader: opera su ogni singolo pixel che viene disegnato e sostanzialmente ne determina il colore finale

# Cenni sulla Pipeline Grafica



# Cenni sulla Pipeline Grafica

Va precisato che il pixel shader attinge i dati anche dalla memoria video per accedere alle informazioni di eventuali texture utilizzate.

# Conclusioni

Le informazioni presentate dovrebbero fornire una panoramica d'insieme necessaria a capire come muoversi quando si fanno videogiochi.

A tale scopo ecco qualche piccolo consiglio per chi vuole cimentarsi in questa branchia:

- Partire sempre dalle basi e con progetti semplici
- Usare componenti (motori grafici, fisici ecc) già fatti per risparmiare tempo ed avere soluzioni ottimali
- Organizzare BENE cosa si vuole fare come farla prima di mettere mano al codice

# Conclusioni

Per domande, chiarimenti, dubbi ecc potete contattarmi all'indirizzo e-mail:

`black_dasa@hotmail.it`