





**intro**

## Chi siamo?

**Valentina Lorè e Nicola Ariutti**

[limulo.net](http://limulo.net) è il nome con cui identifichiamo i nostri interessi e tutto un insieme di progetti che realizziamo.

*limulo.net* è anche il nome del nostro sito internet, all'interno del quale raccogliamo articoli, files, documenti, registrazioni sonore, video e codice di tutti i nostri lavori.



Nasciamo come *sound engineer* (abbiamo entrambi studiato in SAE) e nutriamo parallelamente all'interesse per musica e il sound design una grande passione per la programmazione e la computer science più in generale.

# Di cosa parleremo?

## Parte 1

- Che cos'è il videogioco
- Come si crea un videogioco
- Game Engines

## Parte 2

- Sound Engine tradizionale
- paradigma procedurale
- esempi pratici

pt 1

# **Che cos'è il videogioco?**

**Una breve panoramica storica dell'evoluzione del videogioco**



## Il concetto di gioco

**Narrazione** guidata da **scelte** che uno o più **giocatori** compiono in conformità a un insieme di **regole** che limitano il campo d'azione

- Intrattenimento (non solo)
- Interattività

Nel videogioco in particolare

- Multimedialità



# L'audio nel videogioco

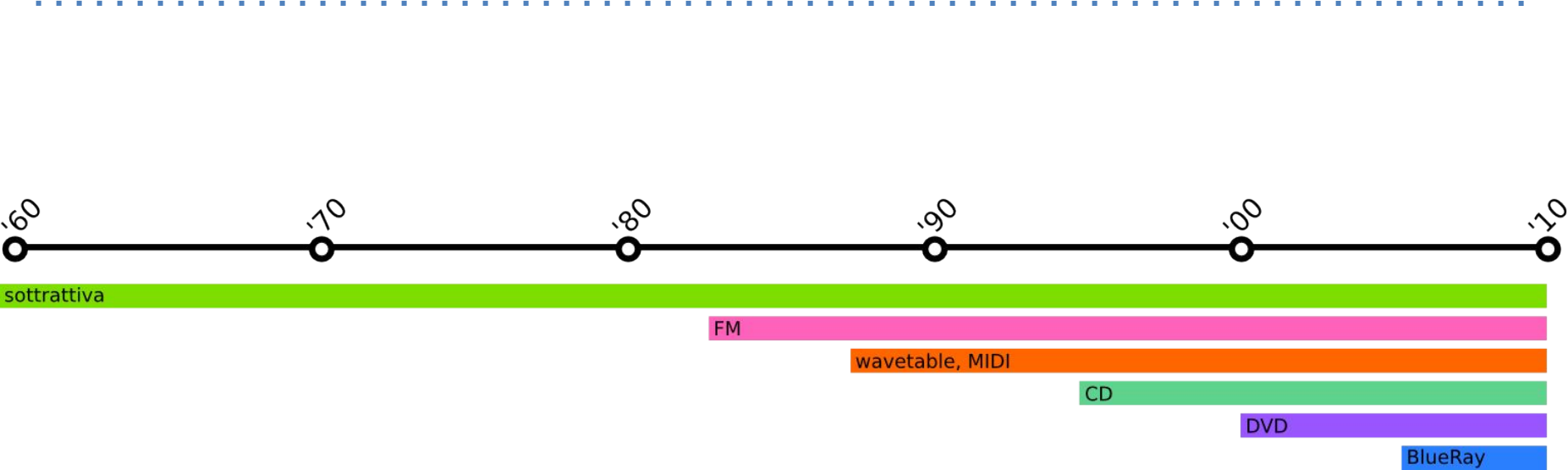
Il videogioco è un software (Input da differenti domini)

Problematiche principali nella programmazione audio per videogiochi:

- gestione delle risorse
- difficoltà di programmazione

Poniamo l'attenzione su

- ruolo dell'audio nel videogioco
- gestione interattiva e dinamica del suono (transizioni)



## Once upon a time...

- 1958: Tennis for Two
- 1962: spacewar!

Supercomputer

Sperimentazione e simulazione



Computer come strumento di gioco da fine anni '70.

## ... intanto in sala giochi...



## La sala giochi



## La sala giochi

Ambiente rumoroso (persone, motori delle macchine)

Duplica funzione del suono: attirare attenzione e coprire rumore ambientale e della macchina

Stile sonoro: rumori forti, squillanti, non continui

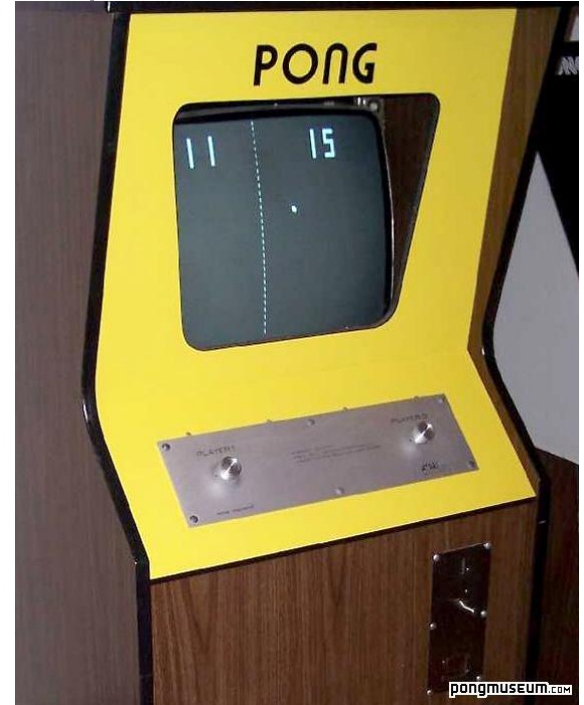
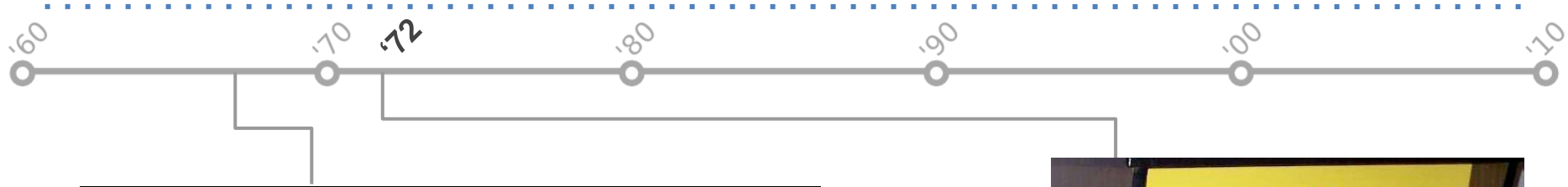
Il suono è slegato dal gioco (vedi slot machine)

## Coin-op: tech specs

Forte componente meccanica

Componenti discreti: difficoltà di programmazione (collegamenti via cavo diretti al chip)

Ogni macchina suono diversamente dalle altre









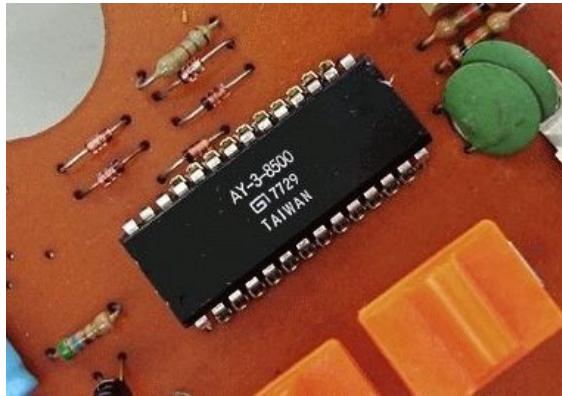
## ... e nel salotto di casa...

Successo con il porting dei giochi arcade

1972: Magnavox Odyssey

1975: [Atari Pong](#) (chip G.I. AY-3-8500 = Pong chip)

Tutti fanno Pong





## VCS

Atari VCS (Atari 2600) con Space Invaders

2 canali + selezione waveform

(tuning dei canali)





## Intellivision

Sistema espandibile

Estensione con nuovo sound chip garantisce  
6 canali audio

# INTELLiVISION.



# Porting del suono

Porting della musica:  
loops delle sequenze sonore

Transizioni tra scene/eventi: hard cuts

Musica e suono più immersivo e non fastidioso per ambiente domestico

## PSG

Inizio anni '80

Programmable Sound Generator (sintetizzatori e DAC)

Speech chip

G.I. AY-3-8910 (arcades. Intellivision, ZX Spectrum, Sega Master System)

Nuove potenzialità per il suono (soprattutto effetti)





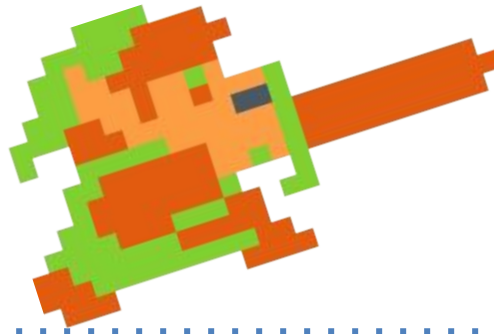
## Maturazione delle consoles

1982: ColecoVision + Donkey Kong

1985: NES + Super Mario

1986: The legend of Zelda (Nintendo)

Inizio indipendenza delle consoles da arcade





## NES

PSG proprietario:

- 2 x pulse
- 1 x tri
- 1 x noise
- 1 DCM (campionatore)







## NES

tecnologia a 8 bit

audio innovativo rispetto predecessori

gestione delle risorse conflittuale

utilizzo di loop per risparmiare memoria



## da 8 a 16 bit

Nintendo abbandona sviluppo per arcade.

Egemonia nel settore rimane a Sega

Sega usa arcade per spingere i suoi prodotti su console



## Sega Genesis

1988/1989: Sega Genesis (Mega Drive)  
a 16 bit

Sintesi FM

musica e effetti hanno pari  
importanza a fini narrativi





## Genesis

PSG 3+1

chip Yamaha per sintesi FM  
(YM-2612)

6 canali + 1 sampler



## Audio nel Genesis

Programmazione difficile: riutilizzo di strumenti per più giochi

### Composizione

loop + trasposizioni + double tracking

### Nuova sensibilità musicale

Uso di nuovi strumenti (violini, fiati, tastiere)

Struttura della composizione (base continua + riff innestati sopra)

Scelta di armonie non occidentali





## SNES

1990: SNES

DSP Sony -> sintesi wavetable

chiptune, estetica a 8 bit

stili musicali in voga  
(dance, hard rock, hip hop)



## Genesis vs SNES

# WHAT NINTENDON'T.

## INTRODUCING MICHAEL JACKSON'S MOONWALKER.



1. In the process, Michael has written the game's original soundtrack.

2. With the classic Moonwalker soundtrack, the tough Moonwalker has the power to use his awesome Moonwalker style.

3. Use Michael's high jumping power to use his awesome Moonwalker style.

4. A high jumping power (jump and control) that the character's unique high jumping abilities.

5. Use the unique power of the Big Moonwalker to move from floor to floor. That's not the same as the other games.

6. All of Michael's famous dance moves are in the game, including Moonwalking and the lean.

7. It's all about the music.

It's here. The hit music video becomes the first ever hit music video game. It was designed by Michael Jackson himself. His moves match his video so closely, it's uncanny. The animation is unmatched. The stereo sound uses Michael's actual voice. And the music is unlike anything you've ever heard before in a video game. It's only possible with the 16-bit power of Genesis.

The totally evil Mr. Big is kidnapping the children of the world. It's up to you to rescue them. In 16 different rounds.

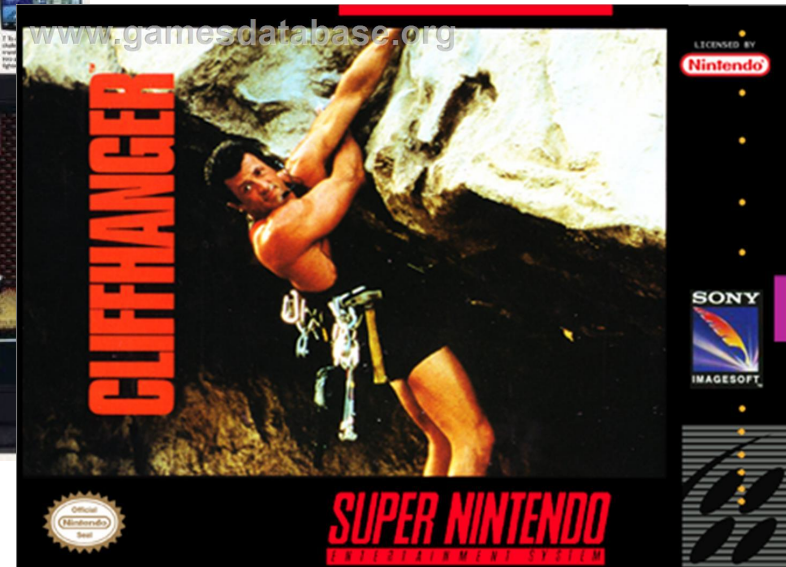
Take on Mr. Big's deadly henchman in Club 30 using your kicks, jump spins, punches and hat attacks. Battle wild dogs and young punks in the high-rise garage. Fend off the zombies in the cemetery. And dodge laser beams in Mr. Big's High Tech Hideout. This game's got it all.

All of this to the stereo sounds of "Smooth Criminal," "Beat It," "Thriller," "Bad" and "Billie Jean."

Finally, you transform into a giant flying robot and use your own laser weapons and heat-seeking missiles.

And from the moment you hit the start button, you know one thing for sure: You can't do this on Nintendo.

1400 x2



www.gamesdatabase.org

LICENSED BY Nintendo

SONY IMAGESOFT

Official Nintendo Seal

**SUPER NINTENDO**  
THE ORIGINAL ACTION ADVENTURE







## Playstation

PS1 : tecnologia CD ROM + cd audio.  
Qualità superiore

perdita di dinamicità con  
l'uso esclusivo di campioni





PS 2: introduce il DVD ROM

Audio: qualità CD superata





PS3: streaming multicanale



## Le consoles verso i computer

L'architettura si complica

Sempre più general purpose

Microsoft Xbox





## Computers

1977: Apple II



Wozniak: *A lot of features of the Apple II went in because I had designed Breakout for Atari. I had designed it in hardware. I wanted to write it in software now. So that was the reason that color was added in first - so that games could be programmed. **I sat down one night and tried to put it into BASIC... I got this ball bouncing around and I said: "Well it needs sound", and I had to add a speaker to Apple II .***



## PCjr

1984: PCjr (IBM)

Sierra: King's Quest

Interfaccia standard per periferiche





## Commodore

Nato con videogame in mente

Emulazione console



# L'audio agli albori del computer

Suono immaturo

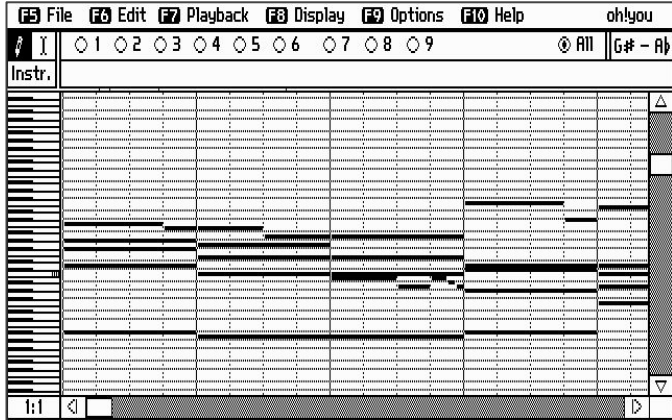
simile a primi arcade (spesso stessi chip)

però il computer è programmabile: molti giocatori diventano creatori e la tecnologia - anche audio - migliora sempre più





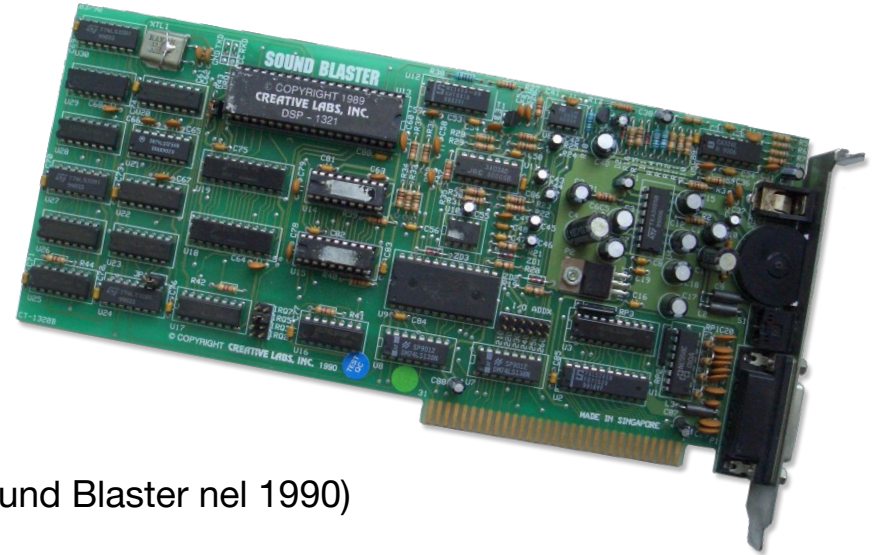
## Schede audio

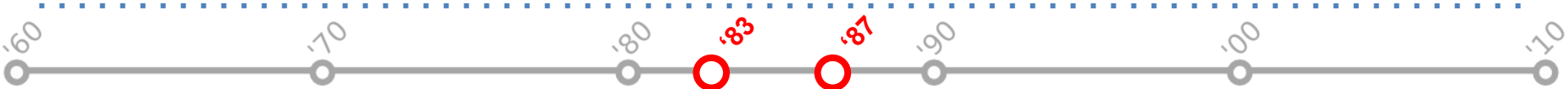


Miglioramento nell'audio

1986: AdLib per Amiga

1988: Creative Technology Game Blaster (Sound Blaster nel 1990)





## MIDI

1983: protocollo trasmissione dati audio creato da Roland

Pro:  
risparmio spazio in memoria  
programmazione semplificata

Contro:  
senza uno standard ci sono moltissime implementazioni differenti

Standard proposti: chip Yamaha e Roland MT32





## GMIDI e GSMIDI

1991: nasce lo standard MIDI e viene ampliato il protocollo in alcune definizioni di strumenti.

Dal 121 al 128 si definiscono effetti sonori (per videogiochi)

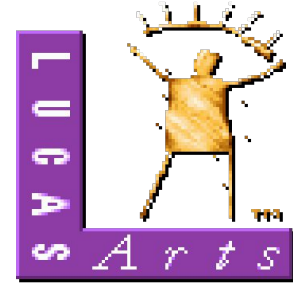
Tecnica compositiva (sequencing) lineare.

Soluzioni creative

110.	Melodic Tom
119.	Synth Drum
120.	Reverse Cymbal
121.	Guitar Fret Noise
122.	Breath Noise
123.	Seashore
124.	Bird Tweet
125.	Telephone Ring
126.	Helicopter
127.	Applause
128.	Gunshot



## iMuse



1991 Michael Land e Peter McConnell sviluppano iMuse, il sound engine di SCUMM, il game engine per le avventure grafiche P&C di LucasArts

Introduzione di componenti audio dinamici tramite invio di messaggi SysEx nei file MIDI.

## iMuse come funziona

2 tipi di messaggi SysEx: markers e hooks

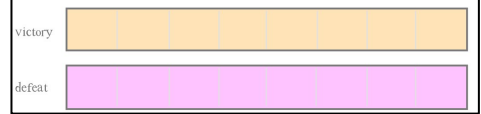
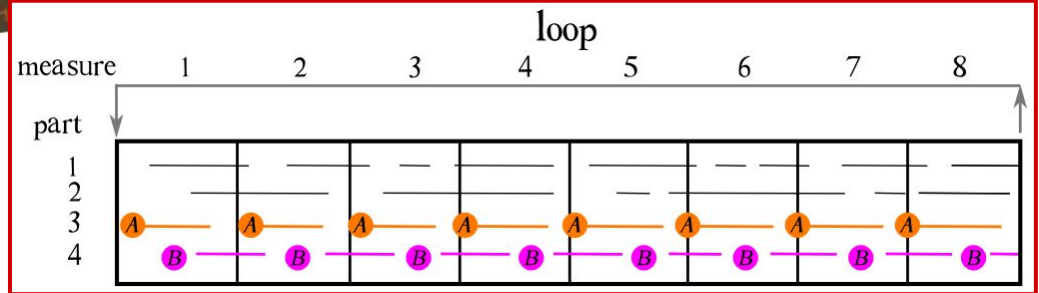
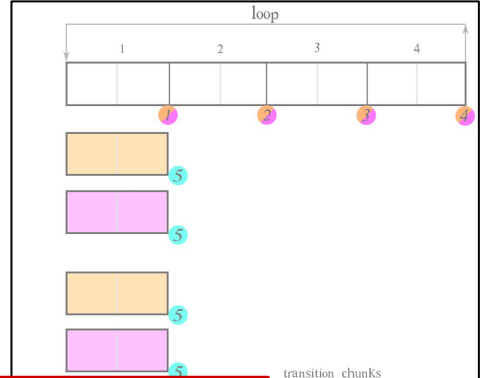
### **markers (ID):**

- posti nel file MIDI nel punto in cui deve essere eseguito un comando
  - il comando è inserito in una coda di comandi nella memoria del sistema
  - quando il lettore MIDI raggiunge il punto, il marker triggera l'esecuzione del comando legato all'ID del marker stesso
- esempio:* fade in/out, variazioni in volume, pause/resume

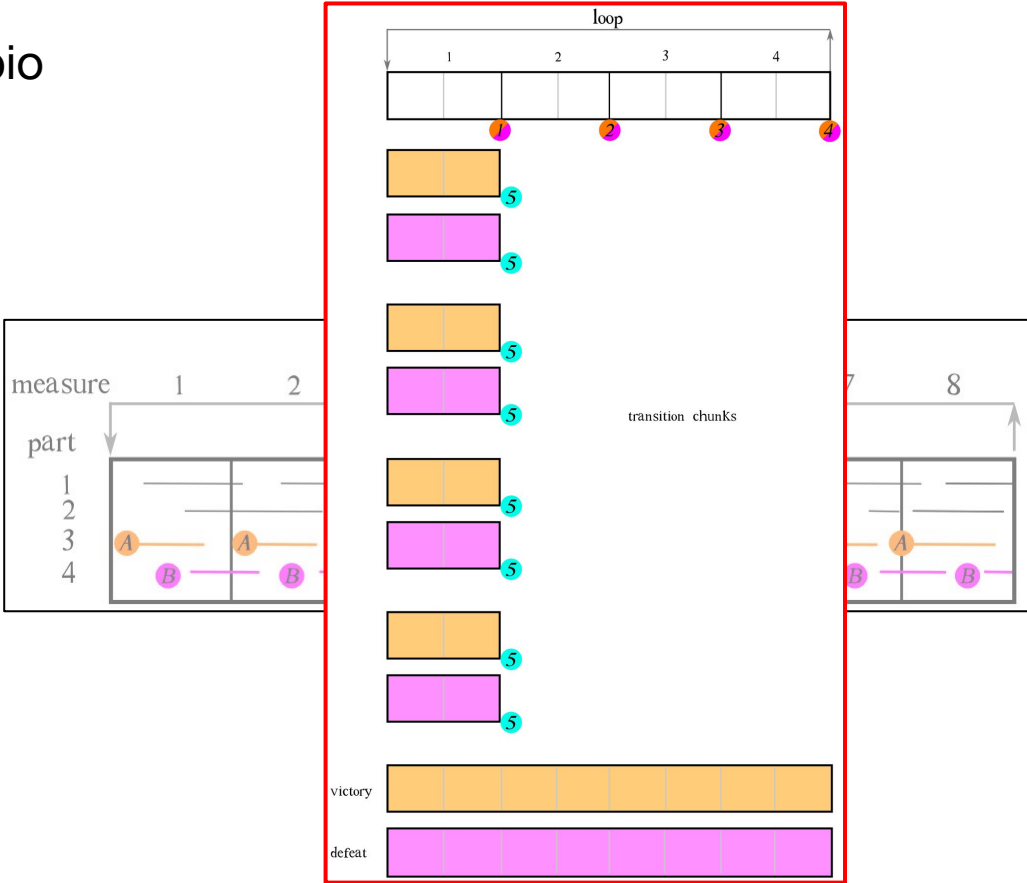
### **hooks (ID + comando)**

- al messaggio si aggiunge non solo un ID ma anche un comando.
  - il sistema resta in ascolto
  - se il lettore MIDI incontra un hook il sistema esegue il comando specificato nello stesso
- esempio:* salti, trasposizioni, abilitazione/disabilitazione di strumenti

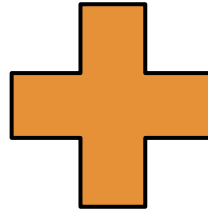
# iMuse esempio



# iMuse esempio



## iMuse: testiamolo

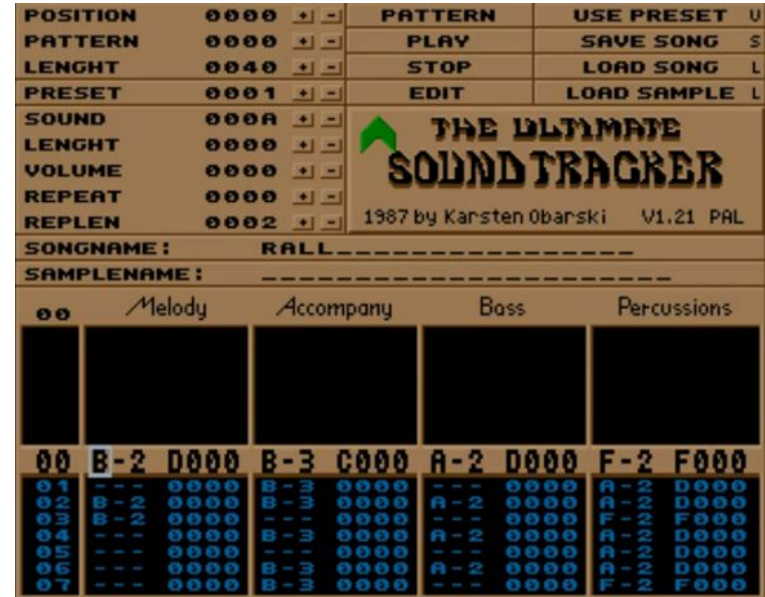




## MOD format

MOD = formato di file  
 sistema modulare inventato da  
 Karsten Obarski per comporre su Amiga

i file MOD si usano all'interno dei **tracker**

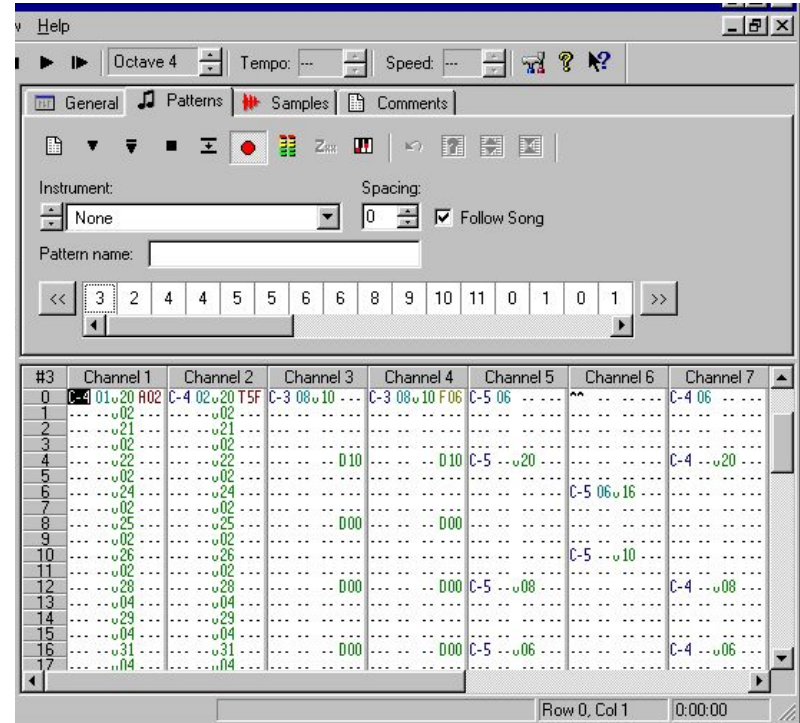


## MOD Format

Si può specificare numero e successione di sequenze, numero di note, strumento associato

Nel file c'è un header che descrive gli strumenti sotto forma di samples

E' possibile programmare dinamicamente usando il MIDI ( pattern per cambiare volume, strumento, ...)

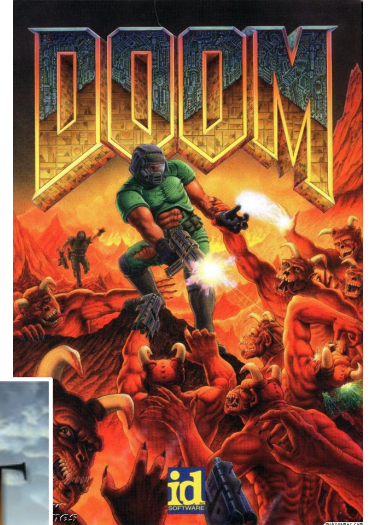


# MOD Format - ancora oggi



## Giochi moderni - un confronto

- Doom (id Software):
  - musica a loop
  - effetti importanti anche dal punto di vista del gameplay
  - non si può mettere in muto l'audio
- Myst (Cyan):
  - musica di accompagnamento
  - musica continua
  - soundscape





## ... e oggi? Nuovi elementi in gioco...

**VR**

**Mobile / handheld  
games**

**Nuovi  
supporti  
(smart TV,  
nuovi OS)**

**iper-realismo**

**Console e PC  
verso unico  
supporto general  
purpose**

## ... e l'audio?

- Surround
- Streaming
- **positioning** per realismo
- samples non dinamici

## ... e l'audio?

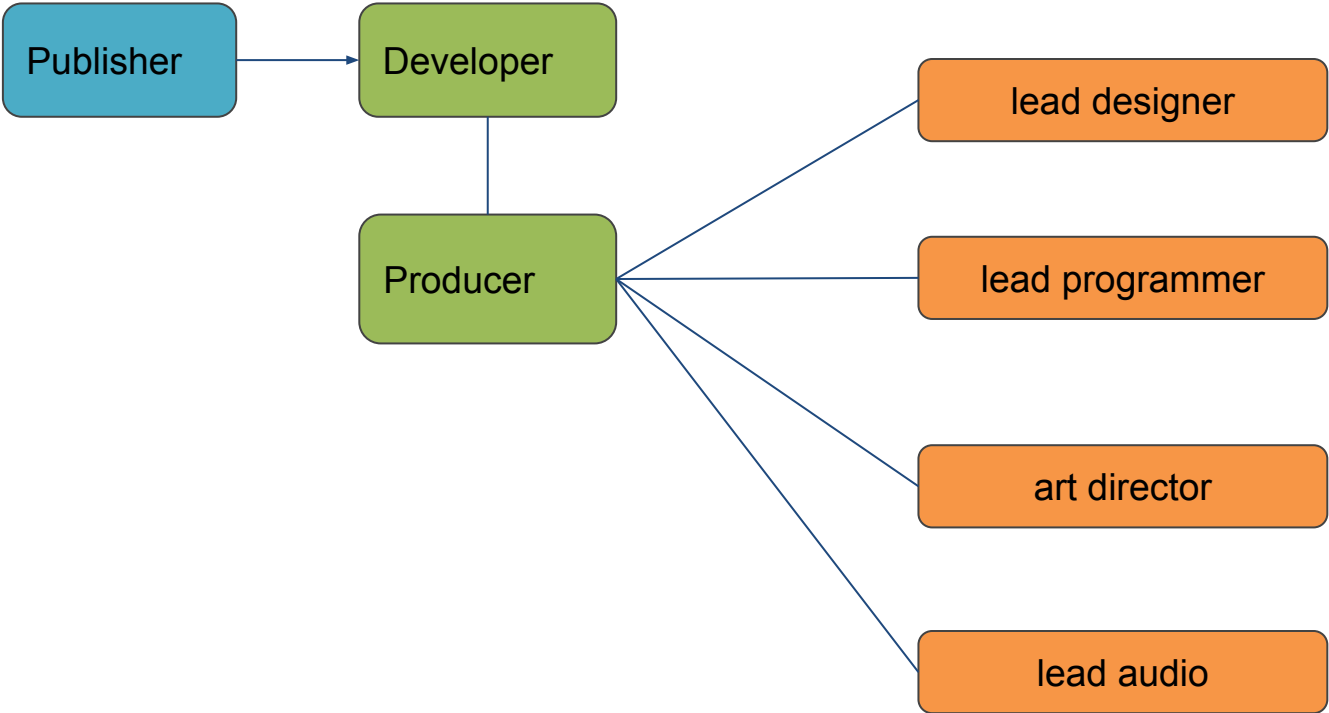
- Surround
- Streaming
- **positioning** per realismo
- samples non dinamici

*e nel futuro???*  
*Che ruolo vogliamo per l'audio  
nei videogiochi?*



# Come si crea un videogioco?

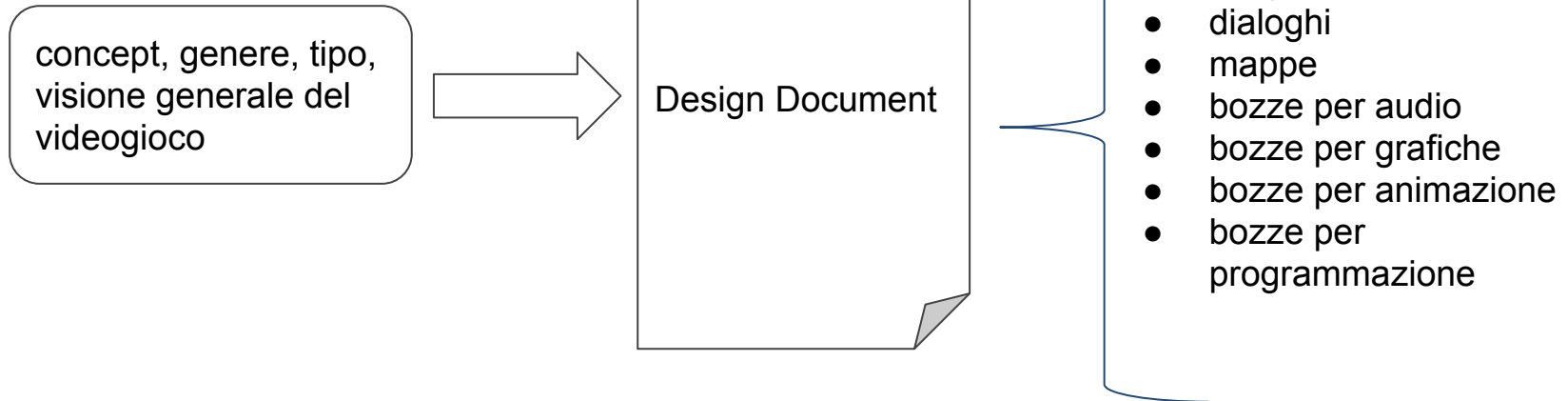
**Le fasi di sviluppo e le figure professionali**



professionals

- 
- 
- 
- 
- 
- 
- 
- 
-

## Pre-produzione



## pre-produzione audio

Musica, Sound FX, Dialoghi

Genere del gioco, ruolo dell'audio nel genere e nel gioco specifico

**cue spotting** (mappa emotiva e mappa dei punti di tensione e rilassamento)

prime idee su effetti, foley, suoni ambientali

elenco di assets da sonorizzare (armi, personaggi, UI, ...)

caratteristiche e limitazioni tecniche della/e piattaforma/e di destinazione

tools da usare (già presenti o da creare da zero): analisi del sound engine



**Audio Design Document**

# Produzione

Ogni punto del design document viene preso in consegna dal rispettivo dipartimento

Ogni dipartimento si interfaccia con gli altri per aggiornarsi sui cambiamenti in corso d'opera

**QA** si assicura che il prodotto sia conforme al design document e alle direttive del publisher

## Produzione audio

viene sistemato e ultimato il cue spotting

viene preparata una **scratch track** di ispirazione per il compositore

si creano foley con l'aiuto di props

layering dei suoni, eq, applicazione effetti DSP

## Produzione audio - il dialogo

### In **voiceover**

professionisti (attori, doppiatori, direttori del doppiaggio)

un attore per più ruoli

recording di singole parti - recording sheet -

studi per ADR

## Produzione audio - il dialogo

Cosa si registra?

- battute durante il gioco
- battute nello script
- parlato durante le cutscenes
- walla
- narratore in voiceover
- cue triggerate da AI

file raw

QA check

DSP FX



## Produzione audio - la localizzazione

Sim-ship o post gold

Non tutti i generi di videogioco necessitano la localizzazione

E' essenziale per i giochi il cui punto di forza è il realismo

Differenze culturali (blending = scrivere per un'altra cultura)

# Produzione audio

Musica, effetti, dialoghi e localizzazione integrati

Uso di strumenti per la modifica in real time - ISACT, Interactive Spatial Audio  
Composition Technology - (FMOD, Wwise, ...) e surround

# Videogame vs Cinema

Linearità vs Interattività

Perdita di dinamica e definizione del suono vs maggior cura del sonoro e della sua definizione

Nel videogioco la post-produzione differisce dalla post-produzione per il cinema

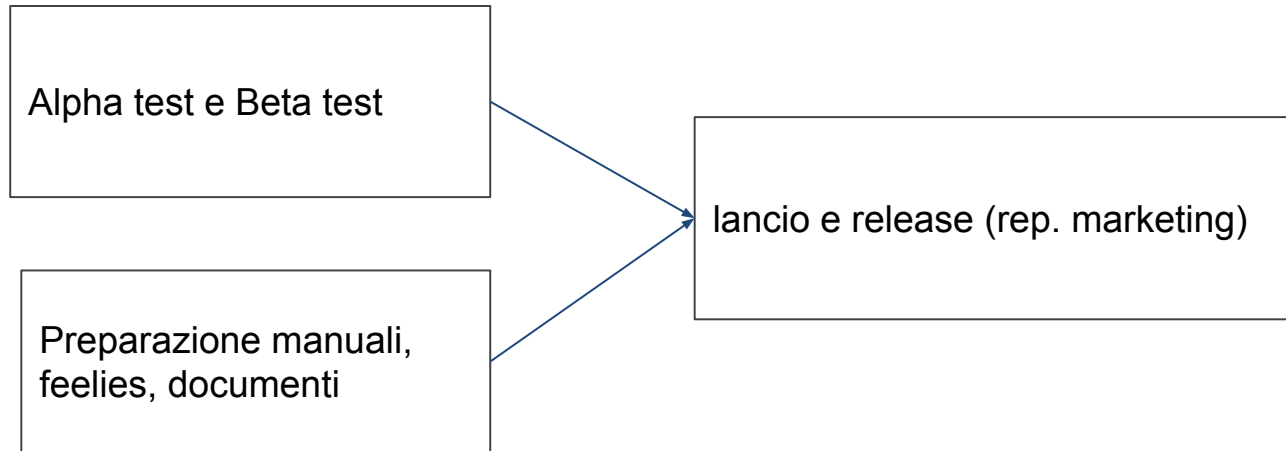
# Post-produzione audio

Mix di sequenze

regolazione dell'audio sul gioco (modifica di effetti, riempimento di buchi, ...)

Uso della psicoacustica per decidere cosa rivelare col suono, come e quando

## Debug & Release



## In realtà

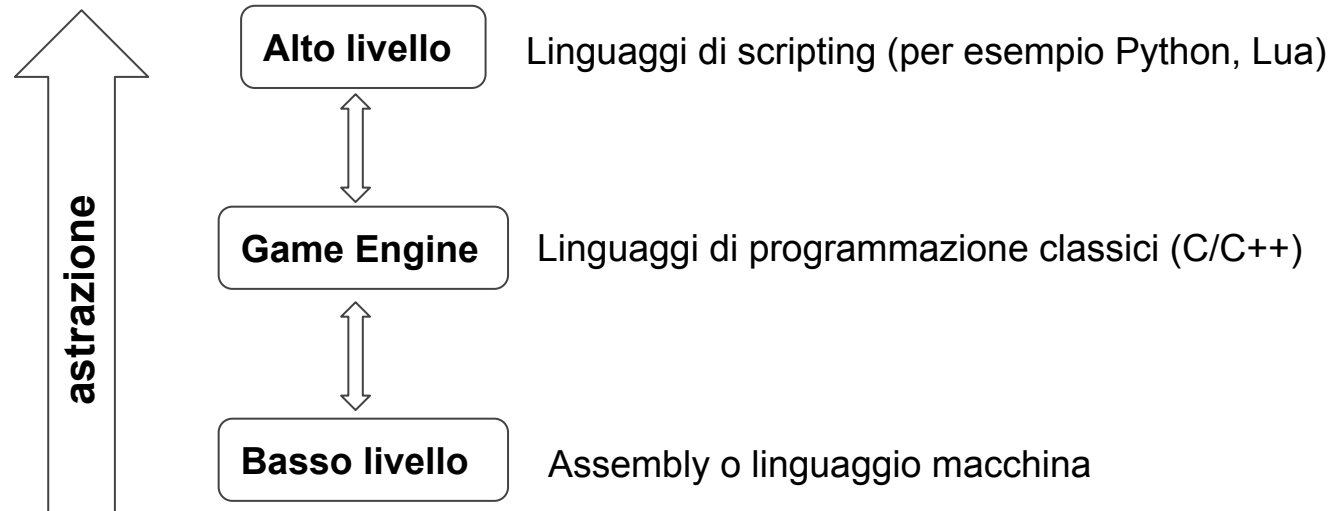
Audio preso in considerazione in fasi avanzate di sviluppo, troppo tardi per avere possibilità di integrazione con le altre componenti

# Game Engines

**I tools di sviluppo moderni, ma non troppo**

## Cosa sono?

**Middleware:** componenti di contatto tra un livello di programmazione ad alto livello (scripting) ed un basso livello (assembly o linguaggio macchina)





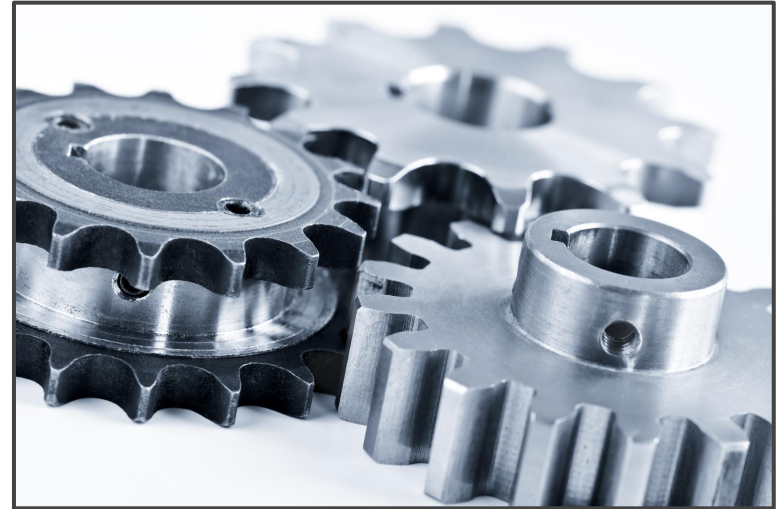
## Quando si utilizzano?

Fase di produzione/sviluppo

Semplificazione del lavoro in team

Astrazione da tecnicismi eccessivi

Lavoro ottimizzato per i programmatori



## Cosa c'è dentro?

- rendering engine (3D/2D)
- physics engine
- interprete del linguaggio di scripting
- AI engine
- animation tools
- sound engine
- memory management
- threading
- supporto video (codec)
- ...



dove li abbiamo già incontrati?



SCUMM engine (1987)



AGI a SCI di Sierra On-line (1984)

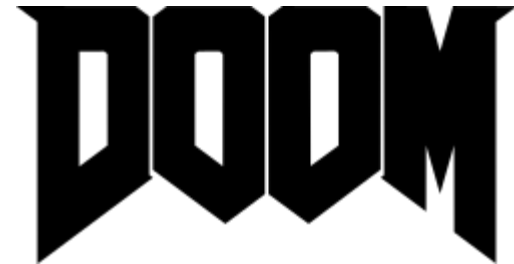
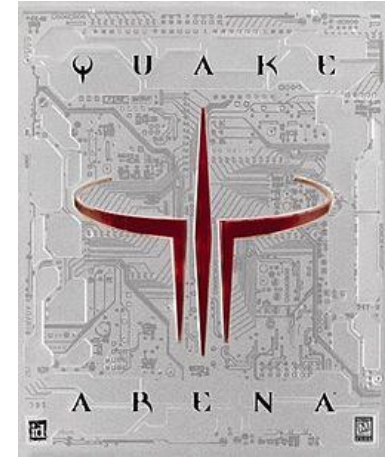


Z-machine di Infocom (1979)

quando nascono i game engine di  
moderna concezione?

Unreal (Epic Games)

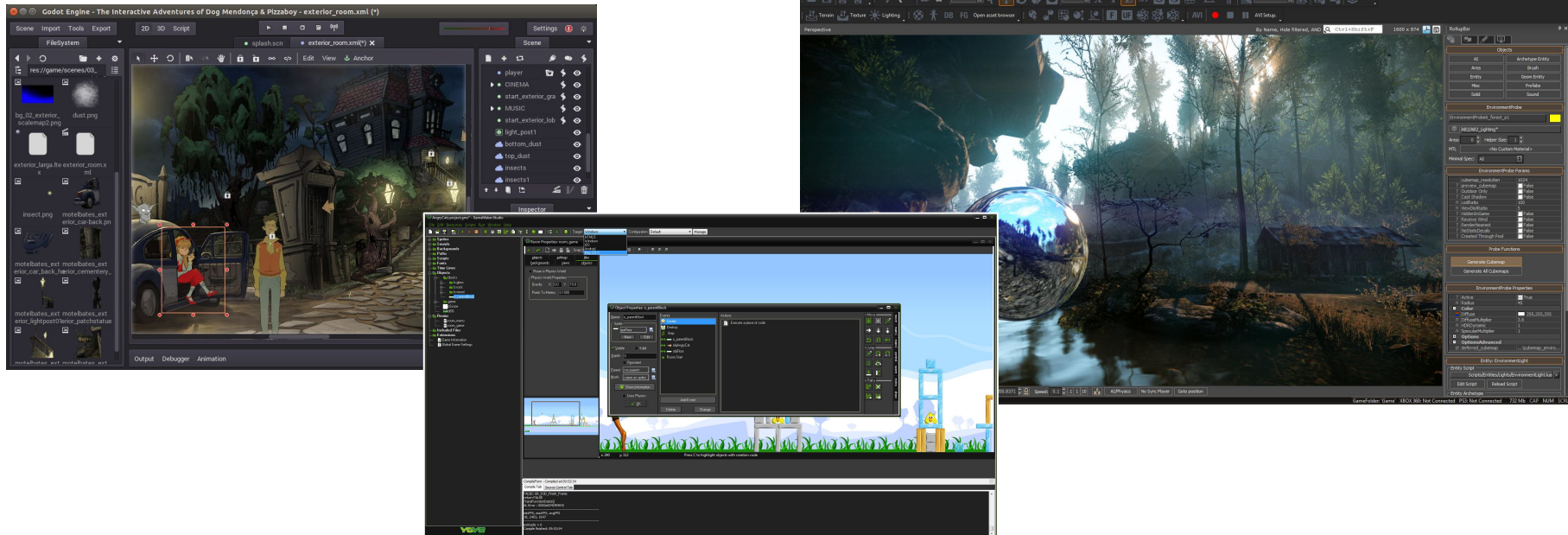
Doom / Quake (id Software)



Doom Engine - 1995  
Unreal engine - 1998  
idTech3 - 1999

# In cosa differiscono dai precedenti?

IDE (Integrated Development Environment): interfaccia grafica di sviluppo



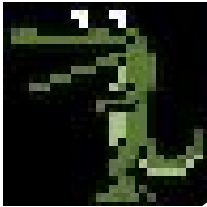
Separazione tra sviluppo di **game contents** (immagini, animazioni, suoni, ...) dalla **meccanica di funzionamento** del gioco.

I team si possono allargare e specializzare

Più spazio dato al level design

Diversi game engine rispondono alle differenti esigenze di sviluppo di differenti tipologie di giochi (FPS, RPG, MMORPG, adventure, platform, ...)

# Un game engine per ogni tipo di gioco



## espansioni

Altri strumenti middleware che estendono alcune funzionalità del game engine

fmod.studio





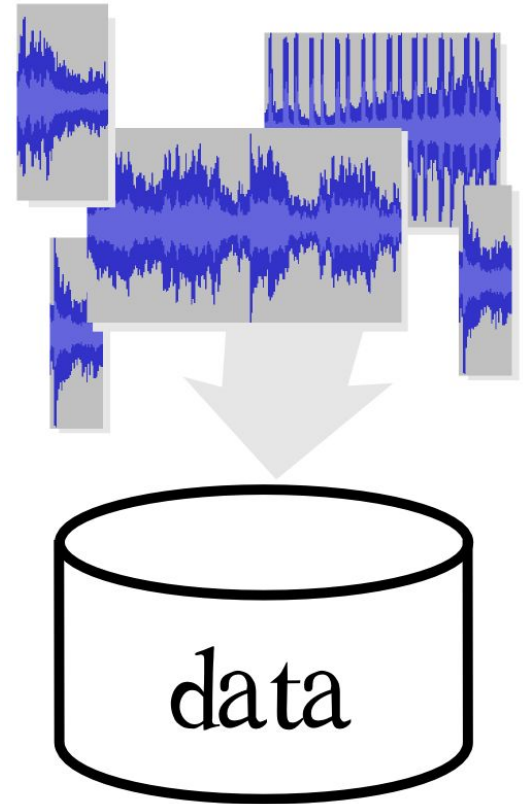
pt 2

## Modello data driven

Nell'ultima parte della sua storia, il suono nel videogioco si presenta come un modello guidato dai dati **data driven model**.

File audio triggerati a seguito del verificarsi di particolari effetti (**event driven system**).

Ai suoni riprodotti si applicano **modificazioni in tempo reale**



## Switching

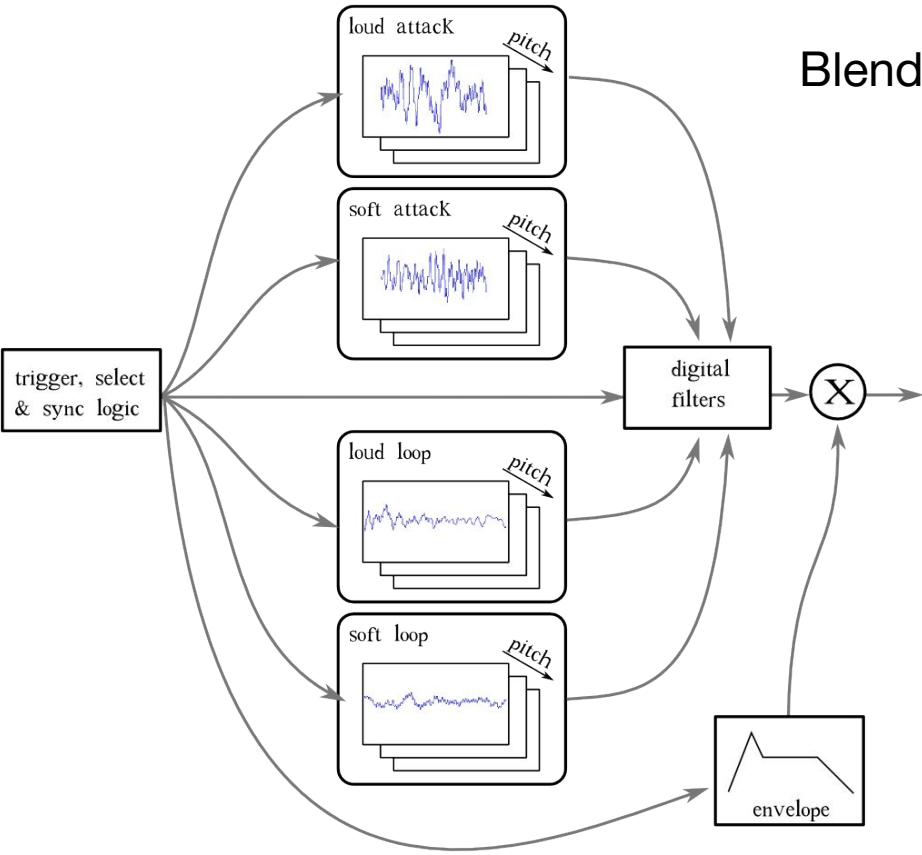
- polyphony;
- voice allocation;
- voice stealing;
- narrazione;



# Random

- **suspension of disbelief**: come spettatori ci lasciamo guidare e ci abbandoniamo alla narrazione.
- un equilibrio difficile da creare a da mantenere;
- un suono ripetitivo viene percepito come **pattern** ricorrente;



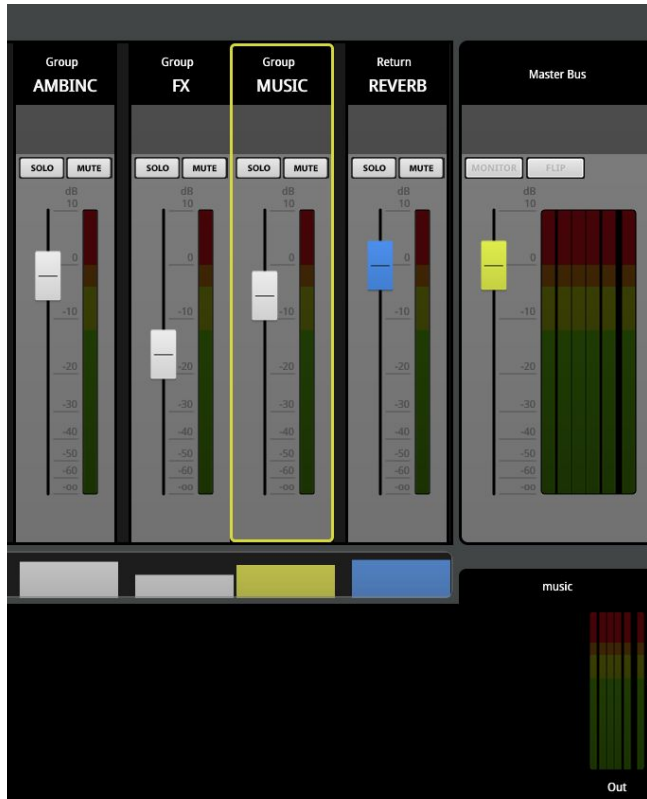


# Blending



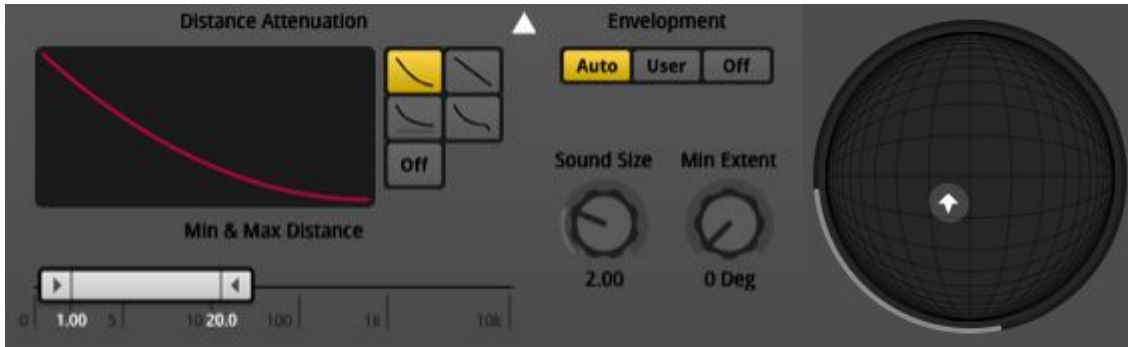
## Mixer, grouping and Buses

riconfigurabile rapidamente (snapshot)



## Real Time Controllers. Posizionamento, Ambianza, Attenuazione

- **listener** ed **emitters**
- riceve dati dal *game engine* e li usa come **parametri**
- **ambiente virtuale**
- attenuazione (occlusione, suono riflesso, materiali)



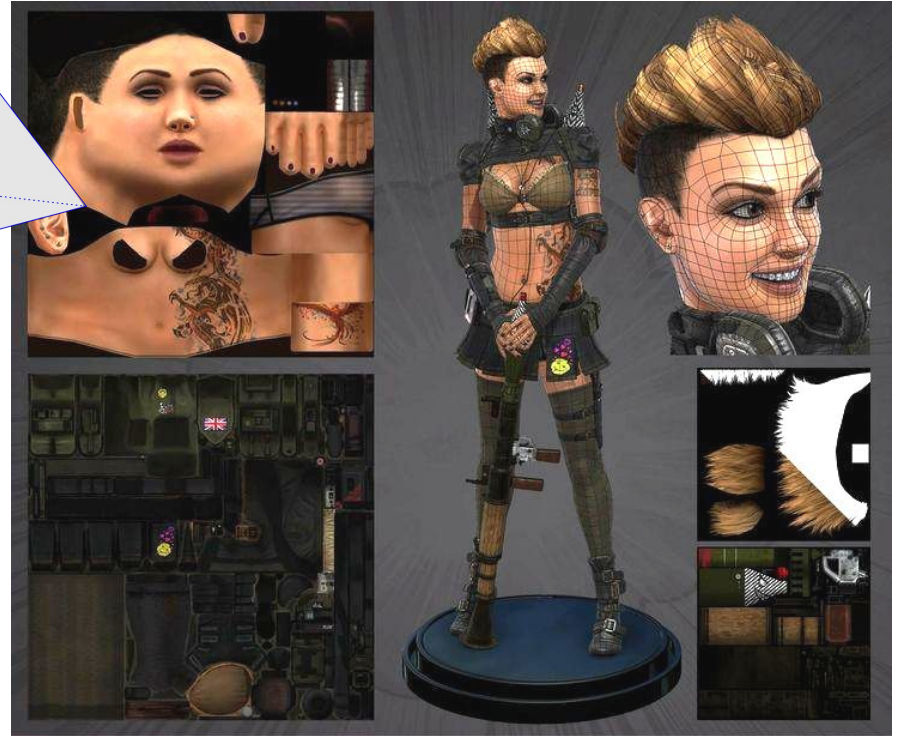
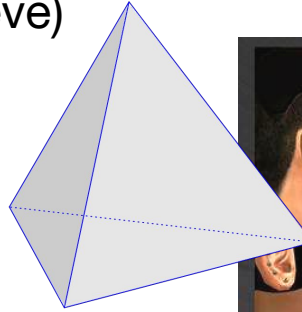
## Dialoghi e Musica





## Come funziona un FPS (in breve)

- mesh (poligoni e vertici);
- textures
- lights
- rendering engine
- particles systems
- world geometry, collisions (physics engine)
- artificial intelligence
- stream di parametri
- 60 fps



## Realismo?

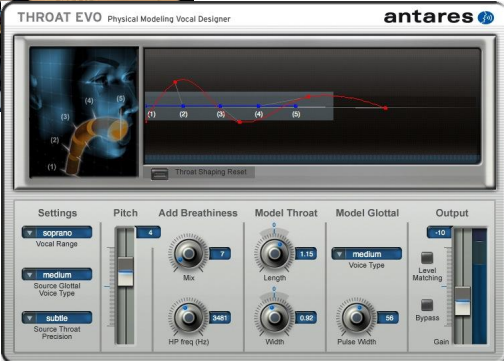
- che cosa è un campione sonoro?
- il suono è importante elemento guida all'interno della narrazione;
- fondamentale componente per la riproduzione della sensazione del realistico.



# Modelling: Virtual Analog

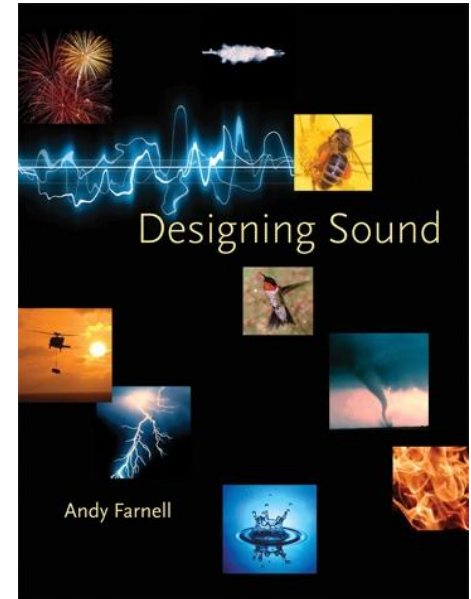


# Modelling: Physical modelling

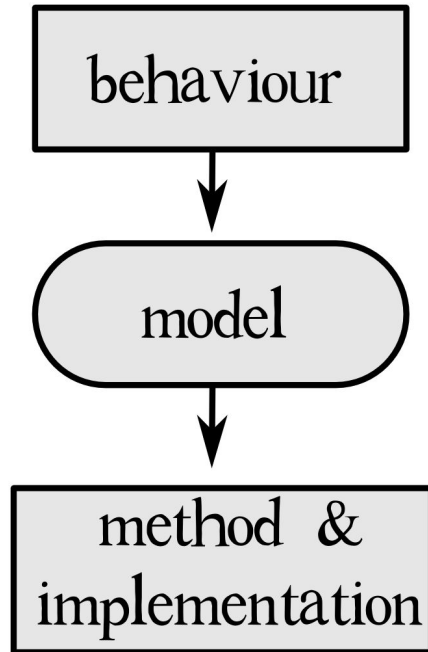


## Suono come processo

- Trasporre per ottenere *sounding objects*
- **sound as a process**



## Suono Procedurale



“**Procedural audio** is non-linear, often synthetic sound, created in real time according to a set of programmatic rules and live input.”

*Andy Farnell*

## Vantaggi:

### Differimento

- recording: decisione prese in anticipo;
- procedurale: molte decisioni sono lasciate al *real-time*;

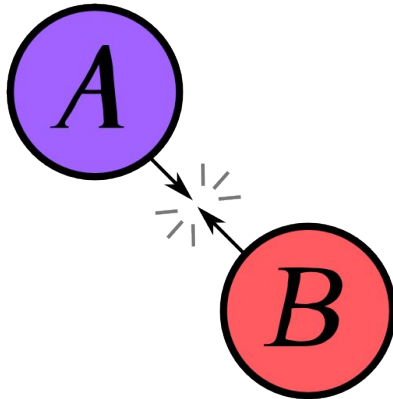
### Variabilità

- modificazione del suono in tempo reale;
- modello completamente parametrico;

## Vantaggi:

### Default forms

- crescita combinatoria
- suono di default associato



### LOAD

- “mipmapping” audio
- modello stratificato
- più layer indipendenti
- psicoacustica





# Svantaggi

- industrial inertia
- new workflows, new skills
- sintesi = falso

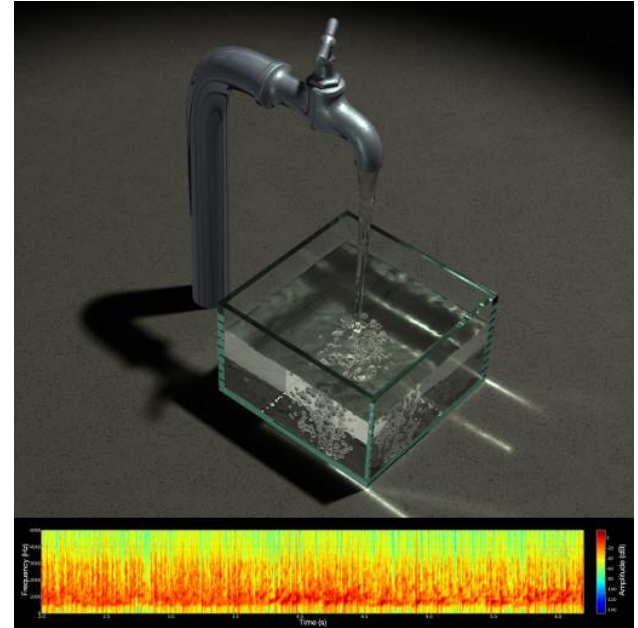
do we really need *realism*?

## Il futuro

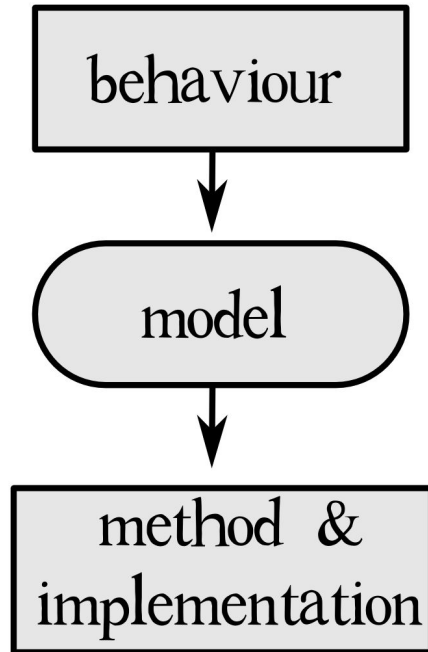
Nella computer graphics: rigging, textures, animazione, modellazione, light, visual fxs

[fracture sound](#), [friction](#), water and [bubbles](#), fire, [crumpling](#), [impatti](#), [acustica ambientale](#)

c'è poi chi si specializza nel processo inverso ([inverse foley](#)) con [risultati sorprendenti](#).



## Esempi: Audio Procedurale, Game Audio Middleware



- Layer di implementazione
- Diversi tipi di linguaggi per l'implementazione
- PureData (Chuck, Supercollider, CSound, etc...)
- fmod

## Dubbi, curiosità, chiarimenti?

Fate un giro sul nostro sito:

- <http://www.limulo.net>

oppure mandateci una mail a:

- [info@limulo.net](mailto:info@limulo.net)

e comunque ci potete trovare anche qua:

- @limulo\_lab
- @nicolaariutti
- @v1a11e1



THANK YOU

follow us @  
[www.limulo.net](http://www.limulo.net)