



INFORMATICA MUSICALE

UNIVERSITA' DEGLI STUDI DI CATANIA
DIPARTIMENTO DI MATEMATICA E INFORMATICA
LAUREA TRIENNALE IN INFORMATICA
A.A. 2019/20
Prof. Filippo L.M. Milotta

ID PROGETTO: 16

TITOLO PROGETTO: Simulatore di propagazione e assorbimento del suono

AUTORE 1: Piccinini Giovanni

Sommario

1.Obiettivi del progetto	2
1.0 Obiettivi generali	2
1.1 Creazione del modello 3D.....	2
1.2 Simulazione di onde sonore	2
1.3 Simulazione delle proprietà dei materiali fonoassorbenti	3
2.Metodo Proposto.....	3
2.1 Studio della modalità di rendering 3D di Processing.....	3
2.2 Studio dei materiali fonoassorbenti di maggiore utilizzo commerciale	5
2.3 Realizzazione del programma in OOP.....	7
3.Risultati Ottenuti	9

1. Obiettivi del progetto

1.0 Obiettivi generali

Lo scopo finale del progetto è la produzione di un software di simulazione tridimensionale utilizzando Processing, un linguaggio di programmazione Object-Oriented. Il programma permetterà di simulare i fenomeni di propagazione, riflessione ed assorbimento delle onde sonore all'interno di una camera insonorizzata. Sarà possibile scegliere la posizione della sorgente sonora all'interno della stanza e la frequenza delle onde sonore, oltre che allo spessore e al materiale di cui sono fatte le pareti.

A questo punto sembra necessario, oltre che utile, soffermarsi su alcuni degli obiettivi principali al fine di permetterne una più chiara comprensione.

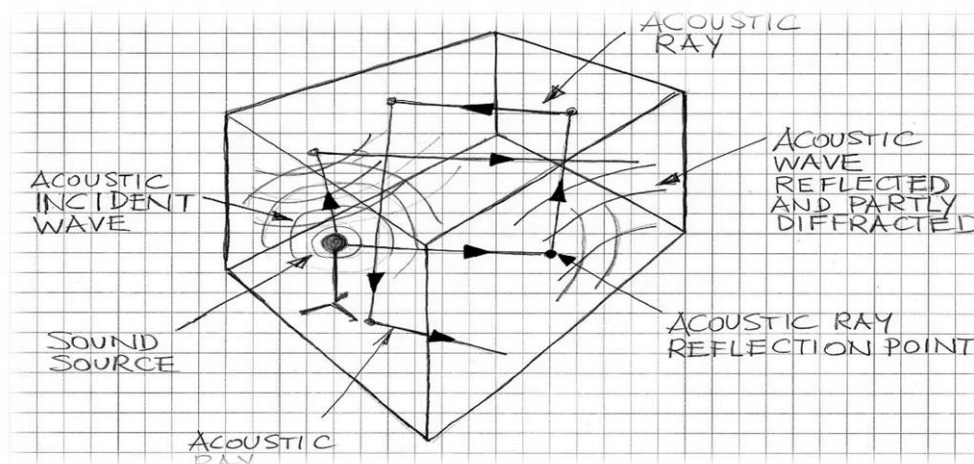
1.1 Creazione del modello 3D

Il software verrà sviluppato interamente su Processing, un ambiente di sviluppo che implementa l'omonimo linguaggio di programmazione OOP. Esso eredita per intero la sintassi e i comandi da Java, aggiungendo inoltre numerose funzioni ad alto livello per gestire in modo semplice gli aspetti grafici e multimediali. Per tale motivo Processing è usato principalmente per lo sviluppo di giochi, animazioni e contenuti interattivi. Maggiori dettagli sulla modalità di rendering utilizzata e sui suoi metodi saranno aggiunti nel capitolo 2.

1.2 Simulazione di onde sonore

Una delle feature fondamentali del programma sarà la capacità di simulare verosimilmente il comportamento di un'onda sonora all'interno di un ambiente chiuso. Ciò significa che :

- Le onde verranno generate da una sorgente puntiforme e si muoveranno ad una velocità fissata nella loro direzione di propagazione.
- Ogni onda avrà una sua intensità prefissata che diminuirà nel tempo, in accordo con la legge dell'inverso del quadrato.
- Quando un'onda incontrerà la superficie di una delle pareti essa verrà riflessa e la direzione risultante dipenderà dall'angolo di incidenza.
- Ad ogni collisione con una parete, una parte dell'intensità dell'onda verrà assorbita. Tale valore dipenderà dal materiale di costruzione della parete e dal suo spessore.



1.3 Simulazione delle proprietà dei materiali fonoassorbenti

Il software avrà la capacità di simulare le proprietà di fonoassorbimento dei materiali utilizzati per la creazione delle pareti della stanza. I quattro materiali studiati nel progetto saranno :

- Lana di vetro con densità di 16-24 kg/m³
- Lana di roccia con densità di 40-140 kg/m³
- Poliuretano espanso
- Polistirolo espanso

Per ogni materiale verrà considerato il coefficiente di assorbimento acustico per le frequenze centrali normalizzate per banda di ottava prese in esame : 125 Hz, 250 Hz, 500 Hz, 1000 Hz, 2000 Hz, 4000 Hz.

2. Metodo Proposto

2.1 Studio della modalità di rendering 3D di processing

Processing offre tre modalità di rendering disponibili fin da subito: **STANDARD** , **P2D** , **P3D** . Mentre le prime due permettono di disegnare solo in due dimensioni , P3D permette di aggiungere una terza asse allo spazio virtuale rendendo possibile la creazione di sketch tridimensionali. P3D risulta quindi essere l'unica modalità adatta allo sviluppo del progetto.

La scelta della render mode avviene all'interno del metodo "setup()" , tramite l'aggiunta alla funzione "size()" di un terzo parametro (in questo caso "**P3D**"). Omettere tale parametro equivale a scegliere la modalità standard di rendering . Il passaggio dello spazio di disegno da bidimensionale a tridimensionale rende necessario l'utilizzo di una nuova coordinata "z" per specificare la posizione degli elementi dello sketch lungo l'asse della profondità, oltre che lungo l'asse x delle ascisse e y delle ordinate. Ciò cambia radicalmente il modo in cui le figure primitive vengono disegnate in quanto funzioni come **rect()** (per il disegno di quadrilateri) o **ellipse()** (per il disegno di ellissi) non permettono di specificare le coordinate lungo la terza dimensione. Ciò rende necessario l'utilizzo della funzione **translate()** che in concomitanza con **pushMatrix()** e **popMatrix()** permette di traslare l'intera matrice tridimensionale che compone la finestra di sketch di Processing e disegnare la figura desiderata, per poi riportare la matrice allo stato iniziale. È anche possibile traslare selettivamente l'asse X, Y o Z utilizzando le tre funzioni **translateX()**, **translateY()** e **translateZ()**. L'esistenza di una terza dimensione rende inoltre possibile la rotazione dei singoli assi attorno a loro stessi tramite l'utilizzo di **rotateX()**, **rotateY** e **rotateZ()**.

Oltre alle classiche figure bidimensionali, P3D offre un set di funzioni per la creazione di figure tridimensionali. È possibile sia costruire figure primitive con **box()** (per il disegno di cubi e parallelepipedi rettangoli) e **sphere()** (per il disegno di sfere) sia creare figure personalizzate grazie alle funzioni **beginShape()**, **endShape()** e **vertex()** che permettono di creare elementi geometrici specificando la posizione di ogni loro vertice. La funzione **box()** prende in input le sue dimensioni, che possono essere specificate da un unico parametro o da un parametro per dimensione. La funzione **sphere()** prende in input il suo raggio mentre **vertex()** prende in input le coordinate xyz che indicano la posizione del vertice.

Una funzionalità molto interessante di P3D consiste nella possibilità di trasformare immagini in **textures** che possono successivamente essere applicate a figure tridimensionali al fine di renderle molto più simili ad oggetti reali. È però necessario fare una distinzione tra le sue due applicazioni : possiamo applicare una texture ad una figura personalizzata o ad una figura primitiva. Nel primo caso basterà aggiungere la

funzione **texture()** tra **beginShape()** ed **endShape()**, dandole in input l'immagine candidata a fare da texture. Nel secondo caso sarà invece necessario dichiarare una variabile **PShape** e salvare al suo interno una figura geometrica primitiva, per poi chiamare su di essa la funzione **.setTexture()** che prende in input un'immagine, la trasforma in texture e la applica automaticamente alla figura selezionata.

P3D permette anche di manipolare la luce dello sketch, simulando gli effetti di diversi tipi di illuminazione nel mondo reale. È possibile definire quattro tipi di illuminazione:

- **ambientLight()** non proviene da una direzione specifica. Tutti gli oggetti sono ugualmente illuminati da tutte le direzioni. Questo tipo di illuminazione viene spesso combinata con altri tipi di luce. È necessario specificare il colore della luce con un colore RGB ed è possibile aggiungere delle coordinate xyz che indichino la posizione della luce,
- **directionalLight()** proviene da una direzione specifica. Il suo effetto risulta tanto più debole quanto più l'angolo di incidenza si avvicina ad un angolo retto. Dopo aver colpito una superficie, la luce si disperde in tutte le direzioni. È necessario specificare il colore della luce con un colore RGB e la sua direzione con un vettore xyz.
- **spotLight()** è simile ad una luce direzionale, ma permette di controllare il suo effetto in modo più specifico. Oltre a colore e direzione, la funzione richiede delle coordinate xyz per indicare la posizione della luce e un angolo che controlla l'apertura del fascio luminoso. Un angolo più piccolo risulterà in una luce altamente concentrata e viceversa un angolo più grande espanderà la zona illuminata, a discapito dell'intensità.
- **pointLight()** è una spotLight con un'apertura del fascio luminoso di 180 gradi. Richiede soltanto un colore RGB ed una posizione.

Quando osserviamo una scena tridimensionale in una finestra di Processing, si può pensare alla visione della scena come se fosse una telecamera. Uno zoom degli oggetti darebbe quindi lo stesso effetto di uno zoom della telecamera e una rotazione attorno alla scena equivarrebbe ad un suo spostamento radiale. Tali effetti possono essere simulati sia tramite l'utilizzo di trasformazioni lineari sia tramite l'utilizzo di una funzione ad hoc: **camera()**. Tale funzione prende in input una serie di parametri che indicano rispettivamente:

- **a, b, c** - Coordinate per la posizione dell'occhio della telecamera.
- **d, e, f** - Coordinate per il centro della scena.
- **g, h, i** - Indica quale asse punterà verso l'alto.

Per modificare la visuale della telecamera a runtime basta rendere variabile qualcuno dei suoi argomenti, come la posizione dell'occhio lungo uno degli assi.

2.2 Studio dei materiali fonoassorbenti di maggiore utilizzo commerciale

Fin dagli albori della civiltà l'uomo ha dovuto affrontare il problema dell'inquinamento acustico. Basti pensare che, a causa dello scalpito dei cavalli sulle strade in pietra, gli antichi Romani furono costretti a bandire i carri di legno dalle strade durante la notte al fine di prevenire disagi ai cittadini. Secoli dopo, nell'Europa Medioevale alcune città decisero di ricoprire molte delle strade principali costruite in pietra con strati di paglia, in un tentativo di ridurre il rumore provocato dal passaggio delle carrozze. In tempi più recenti, durante la Convenzione di Filadelfia, i riformatori della costituzione americana decisero di ricoprire le strade limitrofe con della terra per prevenire che dei rumori esterni li distraessero dal loro importantissimo lavoro.

Al giorno d'oggi il problema dell'inquinamento acustico risulta essere più attuale che mai. Una lunga serie di fattori tra cui la crescita della popolazione, l'urbanizzazione e l'enorme sviluppo tecnologico ha reso necessaria l'invenzione di metodi nuovi ed efficaci per la riduzione ed il controllo dell'intensità sonora. L'applicazione più ovvia delle tecniche di soundproofing è riscontrabile nell'ambito musicale, dove un controllo dell'intensità sonora è assolutamente necessario al fine di registrare accuratamente un qualunque strumento o evitare che un concerto o le prove di un gruppo musicale disturbino la quiete pubblica. Esistono però altre applicazioni al soundproofing meno banali ed altrettanto importanti, se non di più :

- **Contenimento del rumore provocato da grandi macchinari industriali.** Se non trattato adeguatamente esso può portare anche a danni permanenti per chiunque lavori nelle sue vicinanze. Al fine di garantire una maggiore tutela per gli impiegati, sono state create delle leggi per definire la soglia massima di rumore accettabile nell'ambiente di lavoro.
- **Riduzione del rumore di sottofondo all'interno di luoghi pubblici o privati.** Applicare dei pannelli fonoassorbenti in ambienti tipicamente affollati come centri postali o stazioni metropolitane può ridurre significativamente il rumore percepito e il disagio che esso provoca.
- **Riduzione del rumore proveniente da autostrade, strade interurbane e urbane.** A causa del recente aumento del numero di veicoli pro capite, la (discutibile) necessità di muoversi sempre più spesso in auto e l'espansione stradale, il rumore generato dai mezzi di trasporto è diventato una delle maggiori fonti di inquinamento acustico. Per ovviare a tale problema alcune aziende, come la Bosco Italia, hanno ben pensato di creare delle barriere fonoassorbenti ad hoc al fine di assorbirne una parte.

Esistono vari metodi per la riduzione della pressione sonora, materiali diversi offrono performances diverse e la scelta di una soluzione rispetto ad un'altra dipende unicamente dal contesto nel quale dovrà essere applicata. In seguito viene approfondita la tecnica d'interesse per il progetto.

- **SOUNDPROOFING PER ASSORBIMENTO :** Alcuni materiali o oggetti possiedono la capacità intrinseca di assorbire il suono, trasformandone spontaneamente una parte in una piccola quantità di calore. I più utilizzati sono i materiali porosi e gli assorbitori risonanti. Ogni materiale ha una fascia di frequenze entro la quale l'assorbimento è molto efficace, ma allontanandosene le performances peggiorano. Bisogna quindi fare una scelta oculata in base a quali frequenze (o quale range di frequenze) avremo intenzione di far assorbire. I materiali porosi sono efficaci per le frequenze medio-alte. Il loro profilo di assorbimento dipende da un insieme di fattori incluso lo spessore del materiale, la sua densità, la sua porosità e la tortuosità. Gli assorbitori risonanti permettono di assorbire parte della pressione sonora al momento della riflessione. Essi sono maggiormente efficaci per le frequenze medio-basse e il loro range di frequenze è più stretto rispetto a quello dei materiali porosi.

I materiali simulati all'interno del software fanno tutti parte della classe dei materiali porosi e sono stati scelti in quanto, nonostante lo sviluppo di prodotti nuovi ed efficaci, essi rimangono degli "evergreen" nell'ambito :

1. LANA DI VETRO

La lana di vetro è un silicato amorfo ottenuto dal vetro. Essa viene prodotta portando a fusione ad una temperatura molto elevata (1.300-1500 °C) una miscela di vetro e sabbia che viene successivamente convertita in fibre, unite fra loro da un legante. Dopodiché la fibra viene riscaldata a circa 200°C e sottoposta a calandratura. La falda viene quindi tagliata.



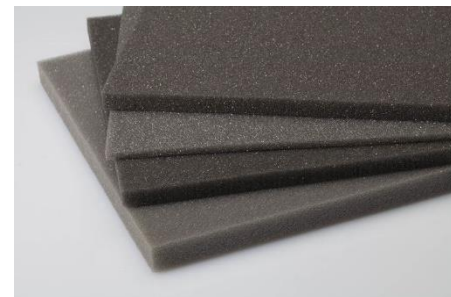
2. LANA DI ROCCIA

È anch'esso un silicato amorfo, ottenuto dalla roccia, ed il processo di produzione è simile a quello della lana di vetro. La roccia basaltica viene fusa ad una temperatura elevata. Viene quindi aggiunto calcare, coke e briquette. Il composto viene convertito in fibre, unite fra loro da un legante. La falda ottenuta viene quindi riscaldata per indurire il legante e tagliata.



3. POLIURETANO ESPANSO

Si ottiene dalla miscela di tre materie prime: poliolo, toluene di isocianato e acqua. Per prima cosa vengono portate ad una temperatura di 20-25 °C. Successivamente le materie vengono dosate all'interno di una testa miscelatrice. La schiuma ottenuta viene quindi versata in una vasca e, attraverso una reazione di polimerizzazione, arriva alle dimensioni finite. Il blocco ottenuto viene quindi tagliato e disposto sull'impianto di maturazione. Dopo 72 ore, la reazione del poliuretano sarà completa e il prodotto sarà pronto ad essere immagazzinato.



4. POLISTIROLO ESPANSO

Si ottiene da perle di polistirene, un derivato industriale del petrolio. Alle perle viene aggiunto il pentano, un idrocarburo presente in natura che funge da gas espandente a causa della bassa temperatura di ebollizione. Mettendo le perle di polistirene a contatto con vapore acqueo il pentano si espande, aumentando la dimensione delle perle fino a 50 volte. Dopo un periodo di attesa che va dai pochi giorni ad una o più settimane, le perle vengono inserite in delle blocchiere dove subiscono nuovamente l'azione del vapore. Durante questo processo le perle si saldano formando un blocco omogeneo. Dopo un periodo di stagionatura, i blocchi verranno tagliati e immagazzinati.



A questo punto, terminata la fase preliminare di studio del progetto, è possibile passare alla fase implementativa.

2.3 Realizzazione del programma in OOP

Il sistema di simulazione è composto da tre elementi fondamentali : la **SORGENTE** , la **STANZA** e l'**ONDA**.A questi tre oggetti reali corrispondono cinque oggetti virtuali.

1. **SORGENTE**

Quest'oggetto si occupa di creare la sorgente sonora. Sarà inizializzato con una terna xyz di coordinate. La funzione **create()** si occupa di creare una sfera di raggio 10 pixel e centro di coordinate xyz che fungerà da sorgente.

2. **STANZA**

Quest'oggetto si occupa di costruire la stanza. Sarà inizializzato con un array di immagini , un valore **i** di indice, un colore **c** in RGB e un valore **t** float. La funzione **create()** si occupa di creare una camera cubica come composizione di figure geometriche ed immagini. Essa avrà dimensione prefissata , pareti di spessore **t** e di colore **c** e userà come texture del pavimento e delle pareti due immagini prese dall'array. Per selezionare la texture delle pareti sarà usato l'indice **i**.

3. **IMPULSO**

Quest'oggetto si occupa di costruire e gestire un singolo impulso di un'onda sonora. Sarà inizializzato con tre terne di valori (una per le coordinate xyz, una per il vettore della direzione e una per l'angolo di rotazione), un colore RGB **c**, un'intensità **i** e il coefficiente di assorbimento **coef**. La funzione **create()** si occupa di creare una curva di Bèzier usando le coordinate xyz e ruotandola di un certo angolo. La funzione **move()** permette alla curva di muoversi lungo la direzione del vettore passato in input. La funzione **energy()** disperde l'energia dell'impulso nel tempo, in accordo con il teorema dell'inverso del quadrato. La funzione **checkEnergy()** controlla se l'intensità è scesa di 60 decibel o più rispetto l'intensità iniziale. La funzione **absorbEnergy()** è usata da **checkBorders()**.Insieme si occupano di rilevare se l'impulso raggiunge una delle pareti e , in caso affermativo , di riflettere l'impulso e assorbitne una percentuale di energia pari a **coef**. La funzione **run()** esegue in serie **energy()** , **create()** e **move()**.

4. **ONDA**

Quest'oggetto si occupa di costruire e gestire un array di oggetti **IMPULSO**. Sarà inizializzato con due terne (una per le coordinate xyz e una per l'angolo di rotazione), una sorgente **s** , un valore **f** che indichi la frequenza, un valore **i** che indichi l'intensità e il coefficiente di assorbimento **coef**. La funzione **create()** permette di generare un array di impulsi usando i parametri in input e controllarne la dimensione. La frequenza con cui gli impulsi vengono generati dipende da **f**. La funzione **checkImpulse()** applica **impulso.checkBorders()** ad ogni impulso dell'array e usa **impulso.checkEnergy()** per verificare che l'energia non sia scesa sotto la soglia. Se ciò avviene , l'impulso viene rimosso dall'array. La funzione **colorCode()** sceglie il colore degli impulsi dell'array in base alla frequenza. La funzione **runImpulses()** esegue **impulso.run()** su ogni impulso dell'array. La funzione **run()** esegue in serie **checkImpulse()**, **runImpulses()** e **create()**.

5. **SUONO**

Quest'oggetto si occupa di simulare l'emissione di suono dalla **SORGENTE**. Sarà inizializzato con una sorgente **s**, un valore **f** che indichi la frequenza, un valore **i** che indichi l'intensità e il coefficiente di assorbimento **coef**. Il costruttore genera un array di oggetti **ONDA** (18 in totale) in modo da simulare l'omnidirezionalità del suono. La funzione **generate()** esegue **onda.run()** su ogni onda dell'array.

La scheda principale del programma è chiamata **PROJECT** e si occupa di gestire gli aspetti più ad alto livello della simulazione. Essa svolge principalmente tre compiti :

- **Caricamento e definizione di dati importanti :**

Nel **setup()** del main vengono specificate le dimensioni della finestra di sketch e caricate le immagini che faranno da textures per le pareti e il pavimento della stanza, oltre che da sfondo della simulazione. Viene inoltre creata una variabile globale di tipo matrice di float che avrà come valori i coefficienti di assorbimento acustico presi dalla tabella sottostante. Per problemi di complessità sono state utilizzate solo le righe in cui il valore “strato d’aria” è uguale a zero.

Tabella 5.4 Valori dei coefficienti di assorbimento acustico, α_{Sab} , di materiali fonoassorbenti porosi.

Materiale	Spessore [mm]	Strato d'aria [mm]	Frequenze centrali delle bande di ottava [Hz]					
			125	250	500	1000	2000	4000
Lana di vetro densità 16-24 kg/m³	25	0	0,10	0,30	0,60	0,70	0,80	0,85
	"	40	0,15	0,40	0,70	0,85	0,90	0,95
	"	100	0,22	0,57	0,83	0,82	0,90	0,90
	"	300	0,65	0,70	0,75	0,80	0,75	0,75
	50	0	0,20	0,65	0,90	0,85	0,80	0,85
	"	40	0,25	0,80	0,95	0,90	0,85	0,90
	"	100	0,45	0,97	0,99	0,85	0,80	0,92
	"	300	0,75	0,85	0,85	0,80	0,80	0,85
	100	0	0,60	0,95	0,95	0,85	0,80	0,90
Lana di roccia densità 40-140 kg/m³	25	0	0,10	0,30	0,70	0,80	0,80	0,85
	"	40	0,20	0,65	0,90	0,85	0,80	0,80
	"	100	0,35	0,65	0,90	0,85	0,85	0,80
	"	300	0,65	0,85	0,85	0,80	0,80	0,85
	50	0	0,20	0,65	0,95	0,90	0,85	0,90
	"	40	0,35	0,85	0,95	0,90	0,85	0,85
	"	100	0,55	0,90	0,95	0,90	0,85	0,85
	"	300	0,75	0,95	0,95	0,85	0,85	0,90
Poliuretano espanso	20	0	0,07	0,20	0,40	0,55	0,70	0,70
	20	40	0,10	0,25	0,60	0,90	0,80	0,85
Polistirolo espanso	25	0	0,04	0,05	0,06	0,14	0,30	0,25

Source : <https://lezioni.strumenti-musicali.info/tecnicodelsuono/2018/03/materiale-fonoassorbente-per-il-trattamento-acustico.html>

- **Gestione della parte interattiva del programma :**

Nel main vengono implementati i comandi che permettono di gestire i vari aspetti della simulazione a runtime : avvio/reset simulazione, cambio angolazione della visuale, cambio frequenza, cambio materiale pareti, cambio spessore pareti , movimento della sorgente sonora.

- **Generazione dello sketch con la funzione “draw()”**

Ed è qui che si trova il cuore della simulazione. Nella funzione **draw()** si trovano le chiamate alle funzioni che si occupano di disegnare e mantenere la **STANZA**, disegnare e mantenere la **SORGENTE** sonora e generare le onde sonore (**SUONO**). Sono inoltre state aggiunte delle righe di codice per la stampa di varie stringhe di testo contenenti sia un tutorial sull'utilizzo della simulazione che alcune informazioni utili : la frequenza delle onde sonore, il coefficiente di assorbimento, il materiale della stanza ed il suo spessore.

3. Risultati Ottenuti

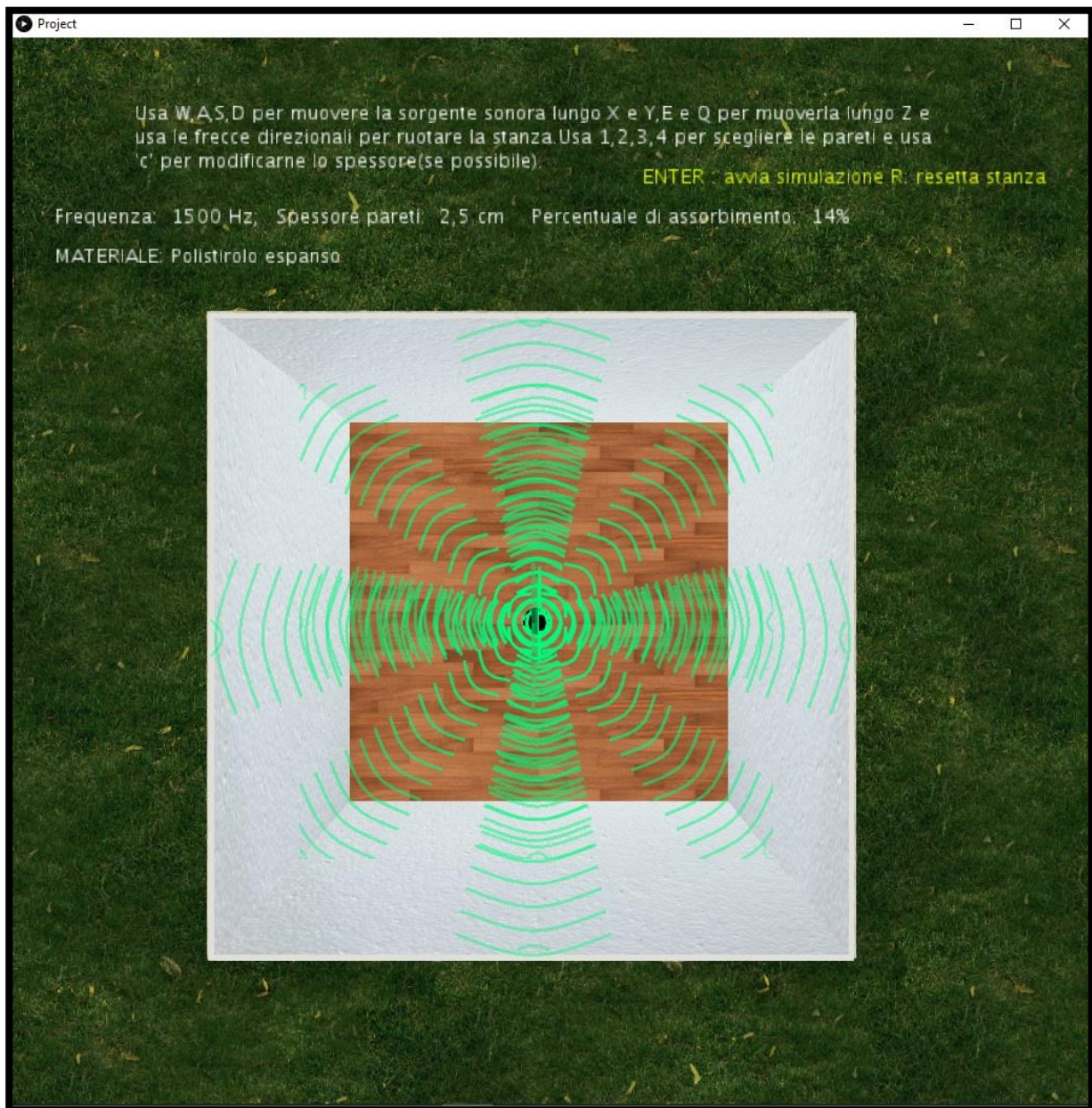
Infine , il programma è pronto ad essere avviato. Questo è ciò che ci si troverà di fronte al momento della sua esecuzione :



Il risultato è una finestra di dimensioni **1000x1000 pixel** con all'interno vari elementi:

- Una lista di istruzioni che permette all'utente di interagire con la simulazione.
- Alcuni dati importanti sulla simulazione. Essi verranno modificati a seguito di specifiche azioni eseguite dall'utente.
- Una camera cubica di dimensione **400 pixel³** dal soffitto trasparente, con pareti in **lana di roccia** di spessore **2,5 cm** e con al centro la sorgente sonora.

È arrivato il momento di cominciare la simulazione vera e propria! Il risultato ottenuto dovrebbe essere simile a questo :



Alla pressione del tasto **ENTER** , la sorgente comincia a generare onde sonore sotto forma di curve di Bèzier con un colore ed una frequenza di creazione che dipendono dalla frequenza in Hertz .

- La frequenza può essere aumentata o diminuita con una precisione di 10 Hz ed ha un valore minimo di 10 Hz e un valore massimo di 4000 Hz.
- Il colore degli impulsi può prendere valori che vanno dal **BLU** per le basse frequenze al **ROSSO** per le alte frequenze , passando dal **VERDE** per le frequenze medie.
- La frequenza di creazione degli impulsi aumenta o diminuisce in accordo con la frequenza e va da un minimo di **0,15 impulso/s** ad un massimo (teorico) di **30 impulso/s**.
- Ogni impulso si muove alla velocità fissa di 60 pixel/s , mentre la distanza tra i suoi due estremi aumenta di 24 pixel/s crescendo da un minimo di 20 ad un massimo di 100 pixel.
- L'impulso aumenta la propria trasparenza al diminuire dell'intensità , fino a sparire del tutto. Questo avviene quando la differenza tra intensità iniziale e attuale supera i 60 dB.

- Al momento della collisione tra un impulso e una delle superfici della camera , la curva viene riflessa con angolo di riflessione opposto all'angolo di incidenza, la distanza tra i suoi estremi ritorna al suo minimo e , se la superficie colpita è di una delle quattro pareti verticali, una parte della sua energia viene assorbita. Tale quantità dipende dalla percentuale di assorbimento , che dipende a sua volta da tre fattori : materiale delle pareti , spessore delle pareti e frequenza delle onde.
- Durante la simulazione è possibile ruotare la visuale sulla stanza ed è possibile muovere la sorgente al suo interno , permettendo all'utente di studiare il rapporto tra posizione della sorgente e distribuzione del suono nello spazio.
- La pressione del tasto di reset riporta la sorgente alla sua posizione iniziale e ferma la creazione di onde sonore ma lascia invariate sia la frequenza che le caratteristiche della stanza (spessore , materiale).
- La pressione del tasto per la modifica dello spessore delle pareti riporta la sorgente alla sua posizione originale e cambia lo spessore delle pareti della stanza (se possibile)
- La pressione di uno dei tasti per la scelta del materiale riporta la sorgente alla sua posizione originale e cambia il materiale delle pareti della stanza.

Sono stati creati diversi range di frequenze utilizzati per selezionare il coefficiente di assorbimento.

<i>Numero</i>	<i>Materiale</i>	<i>Spessore(cm)</i>	<i>0-200 Hz</i>	<i>200-350 Hz</i>	<i>350-750 Hz</i>	<i>750- 1750 Hz</i>	<i>1750- 3500 Hz</i>	<i>3500- 4000 Hz</i>
1	Lana di roccia	2,5	0,10	0,30	0,60	0,70	0,80	0,85
		5,0	0,20	0,65	0,90	0,85	0,80	0,85
		10,0	0,60	0,95	0,95	0,85	0,80	0,90
2	Lana di vetro	2,5	0,10	0,30	0,70	0,80	0,80	0,85
		5,0	0,20	0,65	0,95	0,90	0,85	0,90
3	Poliuretano espanso	2,5	0,07	0,20	0,40	0,55	0,70	0,70
4	Polistirolo espanso	2,5	0,04	0,05	0,06	0,14	0,30	0,25