



AUDIO PROCESSING

UNIVERSITA' DEGLI STUDI DI CATANIA
DIPARTIMENTO DI MATEMATICA E INFORMATICA
LAUREA TRIENNALE IN INFORMATICA
A.A. 2021/22
Prof. Filippo L.M. Milotta

ID PROGETTO: 0B

TITOLO PROGETTO: Analizzatore audio avanzato in 2D

AUTORI: Caziero Kimberly, Gozzo Leandro

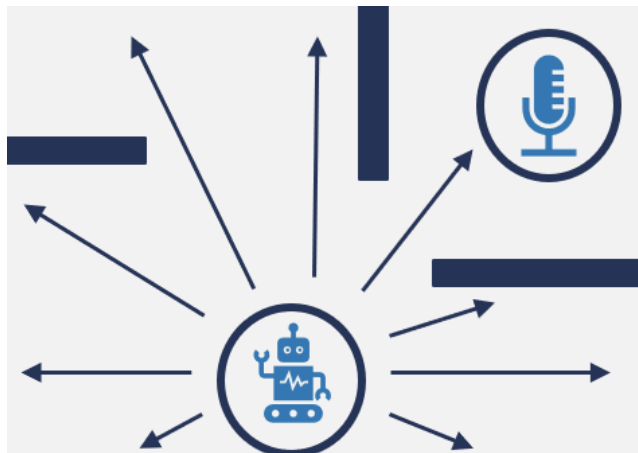
Indice

1. Obiettivi del progetto	2
1.1 Obiettivo generale	2
1.2 Creazione della stanza 2D	2
1.3 Simulazione della diffusione delle onde	2
2 Metodo Proposto	3
2.1 Studio dei materiali presi in esame	3
Gesso	3
Intonaco	3
Legno scanalato	4
Blocchi di cemento	4
Marmo	4
Mattoni	4
Vetro	4
Acciaio	4
PVC morbido e rigido	4
2.2 Progettazione dell'ambiente 2D	5
2.3 Classi utilizzate	5
Materiale	5
Player	6
Onde	6
Diffusione	7
Microfono	7
2.4 Librerie del suono	7
3. Argomenti teorici trattati e risultati attesi	8

1. Obiettivi del progetto

1.1 Obiettivo generale

Lo scopo finale del progetto è quello di ottenere un **simulatore realistico** della propagazione delle onde sonore, creato tramite l'utilizzo di Processing, linguaggio scelto per la facile rappresentazione grafica e per la programmazione orientata agli oggetti. Prenderemo come modello semplificato **la superficie parallela ad un ambiente generico**; all'interno saranno presenti vari oggetti di diverso materiale, che utilizzeremo per osservare i fenomeni di **riflessione** e **rifrazione** delle onde sonore.



1.2 Creazione della stanza 2D

La creazione della stanza è stata codificata con **Processing**, così come gli oggetti all'interno e la sua rappresentazione grafica. Il linguaggio è adatto ad un'interpretazione più immediata del comportamento delle onde; inoltre, potremo far riprodurre dal **monopolo acustico** dei **toni puri** e, tramite un opportuno **microfono** posto all'interno della stanza, capire come il suono verrebbe ricevuto dopo la propagazione.

1.3 Simulazione della diffusione delle onde

Il simulatore si occupa di ricreare un ambiente 2D aperto, diviso in quadrati di lato 1 metro e di ampiezza predefinita, all'interno del quale l'utente potrà:

- **Cambiare il mezzo** principale di propagazione scegliendo tra le opzioni presenti all'avvio, interagendo tramite click;
- **Cambiare specifici riquadri** selezionando uno dei materiali presenti in elenco, sempre dopo l'avvio e tramite click;
- **Spostare Bob**, la sorgente sonora che considereremo puntiforme, tramite le quattro frecce;
- **Spostare il microfono** prima dell'avvio della simulazione;
- **Ascoltare le tracce audio** a disposizione prima della simulazione, e sentire la traccia registrata dal microfono in seguito.

I fenomeni di riflessione, rifrazione e diffrazione verranno studiati tenendo conto delle seguenti caratteristiche:

- La riflessione avverrà quando un'onda cambierà il suo mezzo di propagazione, e di conseguenza, una parte di essa verrà riflessa.
- La rifrazione verrà applicata tenendo conto dell'assorbimento fonico dei materiali contro la quale le onde si scontreranno durante la propagazione.
- La diffrazione dovrebbe avvenire quando un'onda dovrà passare per un'apertura molto più piccola, tuttavia, non è implementabile per via delle specifiche impiegate, spiegate in seguito nel documento stesso.
- La velocità di propagazione dell'onda sonora sarà strettamente legata al mezzo principale di propagazione scelto, che influirà sulla velocità di trasmissione del suono, tenendo conto della densità del mezzo.
- Seguiremo la legge dell'inverso del quadrato per l'intensità dell'onda propagata.

2 Metodo Proposto

2.1 Studio dei materiali presi in esame

Gli oggetti che saranno sparsi per la stanza durante la simulazione fanno riferimento anche allo studio condotto per **sale d'orchestra**, tra le quali: *Teatri Bellini di Catania, Verdi, Pollini, Massimo di Palermo*. I seguenti risultati sull'assorbimento sonoro dei materiali presentati sono stati estrapolati in parte dal documento di **Cammarata, G. (2016) "Acustica Applicata"**, al capitolo 13, sezioni 13.9.4 e 13.9.5, e in parte da studi pubblici sulle proprietà fisiche di alcuni materiali.

Materiali	Frequenze(Hz)					
	125	250	500	1000	2000	4000
Marmo	0.01	0.01	0.02	0.02	0.02	0.03
Mattoni	0.02	0.02	0.03	0.04	0.05	0.07
Vetro	0.03	0.03	0.03	0.03	0.02	0.02
Calcestruzzo intonacato	0.01	0.01	0.01	0.02	0.02	0.03
Cemento	0.36	0.44	0.31	0.29	0.39	0.25
Legno di pino	0.10	0.10	0.10	0.09	0.10	0.12
Lana di roccia	0.26	0.45	0.61	0.72	0.75	0.85
Sabbia	0.24	0.34	0.45	0.62	0.76	0.95

Queste informazioni sono necessarie per il fenomeno della rifrazione, in quanto durante quest'ultimo una parte dell'intensità dell'onda sonora si converte in energia termica e una parte viene deviata nell'ambiente. In una sala d'orchestra, è essenziale scegliere non solo i materiali giusti, ma anche la grandezza corretta della stessa. Si vuole evitare che le armonie prodotte dai musicisti perdano di qualità a causa di fenomeni di **eco nella riflessione**. Al contrario, si favorisce in piccola parte il **riverbero**, che arricchisce la musica.

D'altro canto, è giusto parlare della capacità di assorbimento di certi materiali in relazione al fenomeno dell'**inquinamento acustico**, il quale oggi giorno riguarda non solo le grandi metropoli, ma anche la vita quotidiana di ognuno di noi. Per legge bisogna limitare questo problema a certe soglie in città, ma non solo. Infatti, **i materiali e le tecniche soundproofing** vengono applicate nel **campo industriale** al fine di proteggere l'udito dei lavoratori, soprattutto laddove lavorare a lungo a contatto con macchinari molto rumorosi può risultare nocivo. Senza indugiare oltre, vediamo le caratteristiche principali di alcuni dei materiali sopra citati e di altri ancora, che andremo poi ad utilizzare all'interno del nostro simulatore.

Gesso

Il gesso, come materiale da costruzione, viene estrapolato dall'omonima formazione rocciosa. Il suo utilizzo è legato principalmente a sculture, pareti divisorie fittizie, lastre per l'isolamento termico e fonico, nonché per miscele per intonaci di fondo e intonaci lisci di pareti interne. Da non confondersi con la calce.



Intonaco

L'intonaco è tradizionalmente una malta composta da una parte legante (indurente) che ingloba sabbia granulosa fina. Ci sono molti tipi di intonaco nelle moderne costruzioni in base al legante e alla sabbia di cui è composto, ma il più comune è un composto a secco, detto premiscelato, ed è quest'ultimo che prenderemo in esame.



Legno scanalato

È costituito da legno massiccio a cui sono state apportate delle modifiche tramite macchinari, i quali dotano i pannelli di scanalature che permettono di incrementare l'assorbimento del suono, disperdendo meglio l'intensità delle onde sonore. Nella simulazione supporremo di poter avere delle pareti costituite da quest'ultimo.



Blocchi di cemento

Esistono diversi tipi di cemento, differenti per la composizione, per le proprietà di resistenza e durezza e quindi per la destinazione d'uso. Dal punto di vista chimico si tratta di una miscela di silicati di calcio e alluminati di calcio, ottenuti dalla cottura ad alta temperatura di calcare e argilla oppure di marna, con aggiunte di gesso.



Marmo

Il marmo si forma attraverso un processo metamorfico da rocce sedimentarie. Il colore del marmo dipende dalla presenza di impurità minerali all'interno della roccia sedimentaria originaria. Considereremo un marmo generico al fine delle analisi, in quanto ogni marmo ha proprietà differenti per via della sua composizione specifica.



Mattoni

Forse il primo materiale edilizio nella storia, il mattone di terracotta è molto diffuso nei cantieri, tuttavia oggi i metodi di cottura della terracotta sono molto più raffinati e standardizzati. Il mattone tradizionale è di laterizio, ed è questo che utilizzeremo nella simulazione 2D.



Vetro

È ottenuto tramite la cottura a temperature elevatissime di composti a base di silicio, ossia sabbia. Un esempio di vetro naturale è l'ossidiana, prodotta dal magma vulcanico, ma il vetro può anche formarsi dalla caduta di un fulmine sulla sabbia di una spiaggia. I dati si riferiscono al vetro non temperato.



Acciaio

L'acciaio è una lega ferrosa composta per lo più da ferro e una piccolissima parte di carbonio. Col passare del tempo, le tecniche di produzione dell'acciaio si sono andate perfezionando e diversificando, per cui ai nostri giorni esistono molteplici tipologie di acciai, ciascuna relativa a diverse esigenze progettuali e di mercato.



PVC morbido e rigido

Il PVC, meglio conosciuto come polivinilcloruro, è la materia plastica più utilizzata per la produzione di tubi per edilizia, cavi elettrici, profili per finestra, pavimenti vinilici, pellicola rigida e plastificata per imballi. Ne esistono di diversi tipi in base al materiale da cui si parte nella produzione.

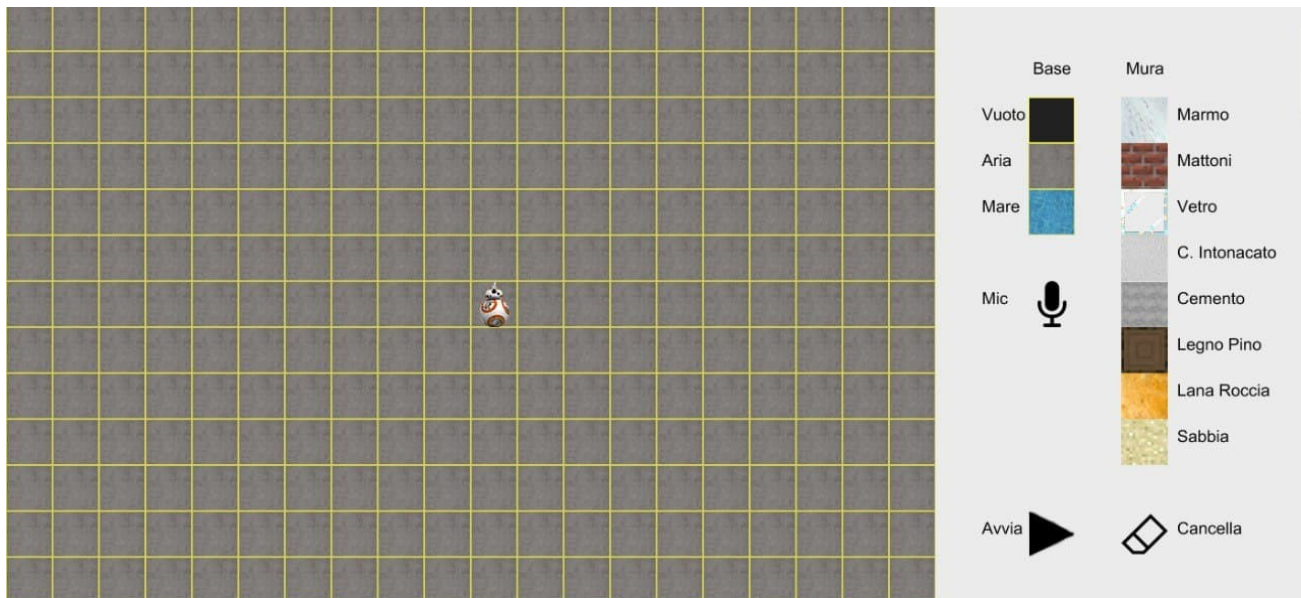


2.2 Progettazione dell'ambiente 2D

La mappa è uno spazio in cui il Player può muoversi liberamente, vincolato solo dalle collisioni con le pareti inserite dall'utente stesso. Nello sketch è presente la costruzione visiva dell'ambiente interattivo e modificabile. Possiamo dividere ciò in diverse sezioni:

- **Mappa:** è una matrice di materiali con dimensione fissa 20x20;
- **Liste di elementi:** sono posti lateralmente, ad ognuno di essi è associato accanto un testo che ne specifica il nome. Sono disposti in maniera analoga a come vengono istanziati negli array opportuni dei materiali (*base* e *mura*).
- **Pulsanti di utilità:** servono per rendere la mappa più funzionale e permettono cancellare un singolo materiale inserito, riportando al mezzo base, di mandare in esecuzione la simulazione e infine di terminarla.
- **Microfono:** è un pulsante che permette di inserire il microfono; se non viene posizionato, la simulazione non verrà avviata.

Qui a seguito un'immagine estrapolata dall'avvio dello sketch, dove sono presenti tutte le funzioni elencate sopra e Bob, il nostro piccolo (50x50 pixel) monopolo acustico.



2.3 Classi utilizzate

Le classi create per la realizzazione del simulatore sono **Materiale**, **Player**, **Onde** e **Microfono**, tutte dotate di metodi costruttori appositi. Inoltre, sarà fondamentale la funzione **Diffusione()** che permetterà il calcolo dell'audio in ogni posizione; sarà collocata nel file principale.

Materiale

La classe **Materiale** è quella che utilizzeremo sia visivamente che a scopo di calcolo per l'audio risultante, e per questo motivo contiene ben nove parametri interni e un array che riguardano entrambi gli aspetti appena citati, ovvero:

- una stringa contenente il nome del materiale;
- un float che rappresenta la celerità del suono all'interno del materiale;
- una PImage 50x50 pixel che sarà la texture del materiale su schermo;
- un boolean che indicherà se il player può passare o meno al suo interno;
- un float per l'intensità;
- due interi che servono per tener traccia delle varie onde percepite;

- due boolean che deriveranno dai precedenti interi; serviranno per tener traccia di eco e riverbero;
- un array che conterrà sei float legati all'assorbimento dei materiali elencati prima.

I materiali saranno ordinati in due array creati nel main, di cui uno serve per selezionare il principale mezzo di propagazione (vuoto, aria, acqua) e l'altro per selezionare il materiale che si vuole collocare su determinati punti della mappa. Come detto in precedenza, sarà presente anche un pulsante **gomma** che permetterà di riportare una casella alla condizione di mezzo di trasmissione base. È chiaro che il materiale "Vuoto" non permetterà la propagazione del suono, ma ci è sembrato interessante inserirlo.

Player

La classe **Player** genererà nel main del codice un solo oggetto, che sarà denominato Bob per comodità. Bob è un **monopolo acustico** dotato dei seguenti attributi:

- due int per la sua posizione nella mappa 2D;
- una PImage 50x50 pixel che sarà il suo segnaposto grafico nella simulazione, anche detta skin.

L'unica funzione al di fuori del costruttore è **isHere()** la quale prende in input due interi per determinare la posizione nella griglia e restituisce un boolean che ci informa se Bob è su quella determinata casella o meno. Il suo utilizzo è legato principalmente all'evitare che l'utente, in fase di preparazione e posizionamento degli ostacoli, ponga un materiale non calpestabile sul Player.

Tramite la funzione specifica di processing chiamata **keyPressed()** è stato possibile implementare il movimento di Bob verso le quattro direzioni corrispondenti ai tasti **LEFT**, **UP**, **DOWN**, **RIGHT**. Inoltre, dalla posizione dell'oggetto, una volta avviata la simulazione, si propagheranno le onde.

Onde

La classe **Onde** è responsabile del comportamento grafico della trasmissione sonora. Per la sua rappresentazione abbiamo optato per istanziare un "cerchio" composto da 45 cerchietti pieni che partono da Bob, i quali andranno inseriti subito in un **ArrayList** designato per la loro gestione. Quest'ultimi saranno dotati di:

- due interi per indicarne la posizione in pixel;
- due float che rappresentano un vettore di movimento per x e y;
- un intero per regolarne l'opacità (dato che sbiadirà man mano che si allontanano);
- un float che ne salverà la distanza percorsa dall'emissione;
- un color che ne caratterizzerà il colore a schermo;
- un float che ne rappresenta la velocità attuale, poiché essa cambierà in base ai materiali.

Il metodo **Disegna()** è quello responsabile dell'apparizione dei cerchietti pieni su schermo. Prende in input l'array delle onde in modo da poterlo modificare con il metodo **Impatto()**. Se non avviene alcun impatto, l'unica modifica è legata ai valori interni dell'onda.

Il metodo **Impatto()** viene richiamato all'interno di **disegna** quando la singola onda va a scontrarsi con un materiale diverso da quello in cui era prima. I parametri passati alla funzione sono un **PVector** che rappresenta il punto dell'impatto, l'array delle onde e due interi che potranno assumere valori +1 e -1, regolanti il movimento sugli assi orizzontale e verticale.

Il **PVector** citato prima viene ricavato tramite la funzione **Intersezione()**, che indicherà dove avverrà il rimbalzo. I parametri presi sono otto interi che rappresentano le coordinate x e y dei quattro punti necessari per calcolare il punto di intersezione.

Gli oggetti verranno inseriti in un **ArrayList** poiché, dopo un certo lasso di tempo, il suono si indebolirà a tal punto da non risultare più udibile. In quel caso verranno eliminati dall'array.

Diffusione

La funzione **Diffusione()** è responsabile della parte di calcolo relativa alla simulazione attiva.

Una volta creato l'ambiente da parte dell'utente, alla pressione del pulsante **AVVIA**, creato appositamente a lato della griglia, non sarà più possibile modificare l'ambiente di simulazione e verrà richiamata la funzione **getStarted()**.

Quest'ultima farà partire otto chiamate alla funzione **Diffusione()**, le quali saranno corrispondenti alle otto caselle che circondano Bob. La funzione è ricorsiva e andrà a chiamare sé stessa ad ogni passo che farà, aumentando un intero che terrà traccia dei passi percorsi dall'onda nel mentre (progressivo).

Le chiamate saranno così gestite:

- Se la posizione della casella successiva è fuori dalla griglia, o l'intensità calcolata è minore della soglia, la funzione ritorna;
- Se il movimento alla casella successiva è in verticale o orizzontale verrà calcolata la funzione per quel movimento e verrà fatto un controllo sulla riflessione, se presente, per richiamarne un'altra nella direzione opposta.
- Se il movimento alla casella successiva è in diagonale, esso sarà richiamato sia in diagonale che nelle sue componenti verticali e orizzontali, dando luogo a tre chiamate ricorsive. Anche in questo caso verrà fatto un controllo sulla riflessione per ognuna di esse, con conseguente richiamo ad altra chiamata, se presente.

Le funzioni ausiliarie richiamate per i controlli sono quattro:

- **Dispersione()**: controlla se l'intensità di un'onda è sufficientemente alta da risultare ancora udibile;
- **fuoriLimiti()**: controlla se l'onda si sta diffondendo ancora all'interno della mappa;
- **controlloRiflessione()**: serve per controllare se c'è un cambio di mezzo tra una casella e la successiva, e richiamare la funzione **Diffusione()** per l'onda riflessa;
- **controlloProgressivo()**: serve per aggiornare i due interi della classe **Materiale** addetti al tracciamento del passaggio delle onde in ogni casella, nonché a modificare alla necessità i boolean di **eco** e **riverbero** nell'oggetto di tipo **Materiale**.

Microfono

La classe **Microfono** è rappresentativa del ricevitore posto nella mappa sia graficamente che praticamente, e perciò sarà dotata dei seguenti attributi:

- due int per indicare la sua posizione;
- un boolean che indica se il microfono è stato già piazzato o meno;
- una **Pimage** per rappresentarlo sulla mappa.

Le funzioni **Set()** e **IsHere()** servono per garantire l'unicità del microfono sulla mappa, e verificare che non ne vengano piazzati altri prima di eliminare il precedente.

2.4 Librerie del suono

La libreria utilizzata per poter ascoltare e modificare l'audio è **processing.sound.***, che ci permette di importare tutti i metodi utili alla realizzazione. Con questa libreria gli oggetti audio saranno di tipo **SoundFile**, il quale prende come parametri un puntatore (generalmente **this**) e un **path relativo all'oggetto**.

È facile riprodurre e interrompere un audio dalla sua riproduzione tramite le funzioni **play()** e **stop()**. La modifica degli attributi del file audio sarà effettuata tramite i metodi **amp()** e **reverb()**.

I **SoundFile** possono essere audio di tipo mp3, wav, AIF/AIFF.

3. Argomenti teorici trattati e risultati attesi

Gli argomenti teorici trattati sono strettamente legati all'**acustica** e ai fenomeni che la interessano. Come ben noto, **la propagazione del suono può avvenire solo se c'è un mezzo di propagazione**, per questo motivo abbiamo aggiunto il **vuoto** tra i mezzi base selezionabili, come controesempio.

Per il resto dei casi possibili abbiamo fatto uno studio dei materiali da prendere in esame e stabilito che, durante la simulazione, potranno essere utilizzati solo dei **toni puri** come suoni di test. Un tono puro è un suono composto da una sola frequenza ed è periodico.

Nonostante siano per lo più impossibili da ritrovare in natura, la scelta di operare con essi è stata condizionata dal fatto che i dati dell'assorbenza sonora, relativi ai materiali, sono stati ricavati tramite test con toni puri, ed in questo modo sarà possibile avere il massimo dell'attendibilità.

Quando un suono si propaga in un ambiente devono essere fatte varie considerazioni: **a che altitudine siamo? Che temperatura c'è? Da cosa è composto il mezzo?** Per semplicità noi abbiamo fatto riferimento ai nostri mezzi principali di trasmissione, eccetto il vuoto, con dei parametri di riferimento fissati. L'aria è fissata a 20° con un'atmosfera di pressione, mentre per l'acqua abbiamo considerato quella marina a 20° con salinità media.

Quando poniamo degli ostacoli nel nostro ambiente di riferimento, ciò che accadrà è una modifica al modo in cui il suono si propaga, che essa sia un "rimbalzo" sull'oggetto, un passare attraverso o l'amplificarsi nell'ambiente, o ancora la sua scomparsa in certi punti. Questi eventi prendono il nome di **fenomeni di deviazione delle onde sonore**.

Tra queste ritroviamo la **riflessione**, la quale può dar origine all'eco o al riverbero, in base alla nostra distanza dalla parte che riflette l'onda e al mezzo in cui ci troviamo.

A seguire la **rifrazione**, che avviene quando cambia la temperatura o il mezzo di propagazione, e tende a far "salire" o "scendere" le onde sonore. Mostriamo questo fenomeno con le nostre onde che si "duplicano" con intensità minore.

Infine, la **diffrazione** si origina quando un'onda ha una lunghezza d'onda molto più grande dell'ostacolo (o fenditura) che incontra; per i parametri presi in esame sarà impossibile ricreare questo fenomeno, in quanto l'apertura più piccola avrebbe comunque un metro di ampiezza.

Per la scelta di operare con toni puri abbiamo ritenuto opportuno fare a meno del dominio delle frequenze, limitandoci ad **operare sui dati del dominio del tempo**. Alla fine della programmazione del simulatore ci si aspetta che esso sia in grado di ricreare il comportamento della propagazione del suono in un ambiente, aperto o chiuso che sia, nel quale sono stati posti ostacoli di vario materiale.

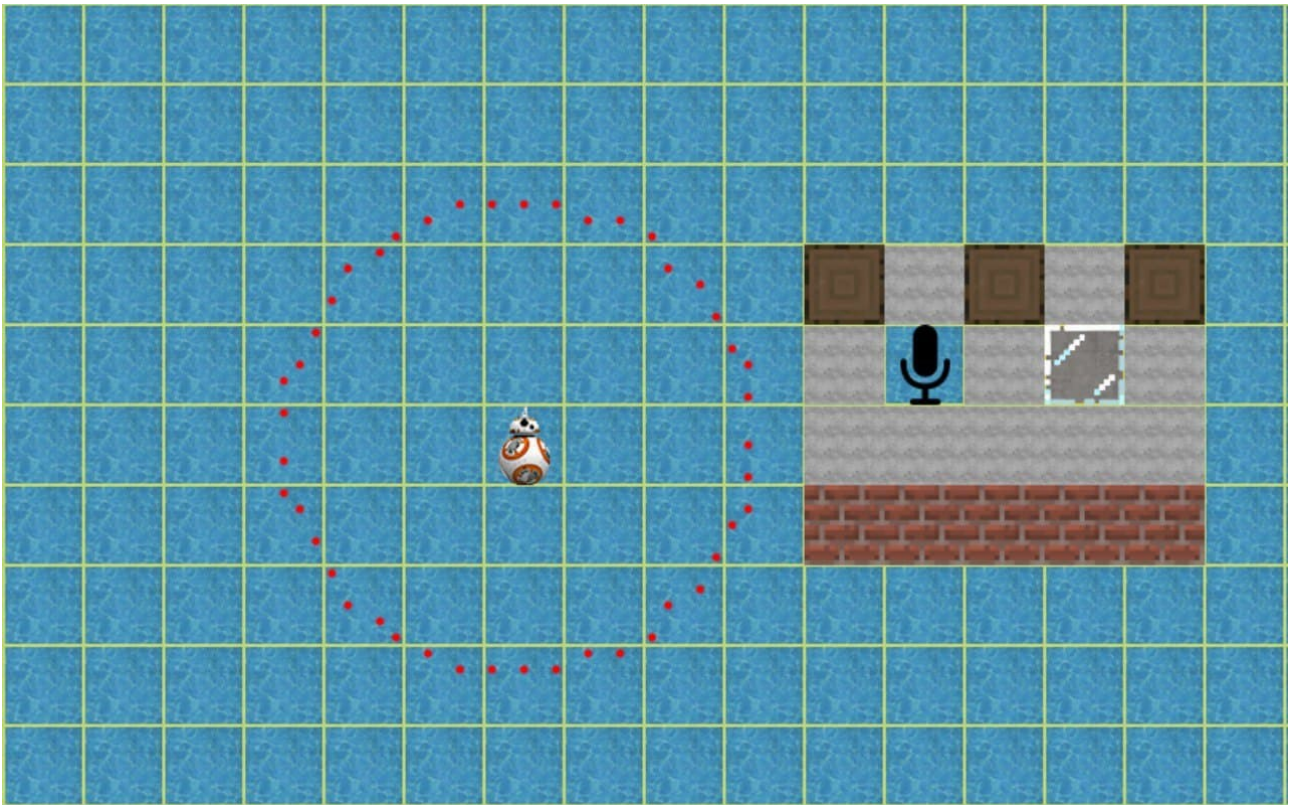
Tra le varie problematiche rimaste da eliminare o migliorare all'interno del codice del simulatore ve ne sono diverse, più o meno gravi, da attenzionare:

Bug minori noti

- Prima di poter selezionare e piazzare il microfono, deve essere stata selezionata una parete;
- La propagazione grafica delle onde non è "circolare" (possibile difetto di conversioni da float o int e viceversa);
- In alcuni casi del calcolo audio, se dei valori scendono sotto la SOGLIA (vedi codice per maggiori informazioni), si genera un loop apparentemente infinito, tuttavia non dannoso ai fini dell'output del microfono virtuale.

Bug maggiori noti

- La propagazione grafica delle onde è corretta, ma non rimbalzano né vengono assorbite dalle pareti quando dovrebbero, limitandosi a stallare nell'ultima posizione dopo il rimbalzo.
- L'intensità audio in caselle non verticali o orizzontali rispetto a quella di propagazione risulta anomala. La causa potrebbe essere legata a qualche errore di calcolo (o caso imprevisto) nelle funzioni che modificano l'intensità, come per esempio la legge dell'inverso del quadrato.



Non escludiamo la possibilità di riprendere in futuro il progetto, completandolo e migliorandolo, per questo motivo sono ben accetti consigli e aiuti. Il codice è un open-source che chiunque può visionare al link github qui sotto:

<https://github.com/Leanir/Advanced2DAudioAnalyzer>