

Digital Audio Workstation (DAW) Multitrack cu efecte simple

Jireadă Teodor
Grupa 1409A

I. Descrierea tematicii proiectului

Proiectul se încadrează în domeniul Multimedia Systems Development și al procesării digitale a semnalelor (DSP), având ca scop realizarea unei aplicații desktop de tip Digital Audio Workstation (DAW).

Editoarele audio sunt instrumente complexe, costisitoare și dificil de utilizat pentru începători. Tematica aleasă propune dezvoltarea unei soluții software accesibile, care îmbină manipularea fluxurilor de date în timp real cu o interfață grafică interactivă.

Proiectul abordează provocarea de a gestiona redarea audio cu latență scăzută într-un limbaj interpretat (Python), necesitatea sincronizării precise între procesele de backend (engine-ul audio) și interfață grafică, precum și implementarea unor algoritmi matematici pentru efecte audio (filtrare, distorsiune, delay). De asemenea, proiectul pune accent pe arhitectura software, utilizând Design Patterns consacrate pentru a asigura extensibilitatea și stabilitatea aplicației.

II. Obiectivele proiectului

Obiectivul general a fost crearea unei aplicații funcționale, modulare, capabile să importe, să editeze, să proceseze și să mixeze fișiere audio multiple într-un mediu vizual intuitiv.

Obiective specifice:

1. Dezvoltarea unui motor audio performant:

- Implementarea unui sistem de mixare în timp real a mai multor canale audio, folosind biblioteci optimizate numeric (NumPy, SoundDevice).
- Gestionarea bufferelor audio pentru a asigura o redare fluidă, fără întreruperi, menținând o latență perceptibilă minimă.

2. Implementarea algoritmilor de procesare a semnalului (DSP):

- Dezvoltarea unui sistem de efecte modular (Effect Rack).
- Implementarea matematică a unor efecte audio standard:
 - EQ (Egalizator): Filtrare de frecvențe.
 - Delay: Crearea de ecouri prin manipularea bufferelor circulare.
 - Distortion: Modificarea neliniară a amplitudinii semnalului (Hard/Soft Clipping).

3. Arhitectură software:

- Utilizarea arhitecturii modulare pentru separarea logicii de business (core) de interfață grafică (ui).
- Implementarea pattern-ului Command pentru a gestiona un sistem complex de Undo/Redo multinivel, permitând utilizatorului să revină asupra modificărilor făcute în proiect.

4. Interfață grafică avansată (GUI):

- Crearea de widget-uri custom pentru vizualizarea formelor de undă și a parametrilor de efecte (Knobs/Sliders).
- Sincronizarea vizuală a Timeline-ului cu poziția curentă de redare (Playhead).

5. Persistența Datelor (Project Management):

- Proiectarea unui format de fișier propriu (bazat pe structura JSON) pentru salvarea și încărcarea proiectelor.
- Serializarea stării complete a aplicației: căile fișierelor, setările de volum/pan, lanțul de efecte activ și parametrii acestora.

III. Descrierea tehnică

Această secțiune detaliază soluțiile software adoptate, structura internă a codului și algoritmii utilizati pentru realizarea funcționalităților de procesare audio și interacțiune cu utilizatorul.

1. Arhitectura Sistemului

Aplicația este construită pe o arhitectură modulară, urmând principiul Separation of Concerns (SoC) și adaptând modelul arhitectural Model-View-Controller (MVC). Codul este organizat în două pachete principale:

- Core (Model & Controller Logic): Conține logica de business, algoritmii de procesare a semnalului (DSP), gestionarea stării proiectului și motorul audio. Acest strat este independent de interfață grafică.
- UI (View): Gestionează afișarea elementelor grafice folosind biblioteca PySide6 (Qt), captarea evenimentelor de la utilizator (mouse, tastatură) și randarea vizuală a formelor de undă.

2. Motorul Audio (Audio Engine)

Componenta centrală a aplicației, AudioEngine, funcționează pe baza bibliotecii “sounddevice”, care oferă acces la API-ul audio al sistemului de operare (PortAudio):

- Procesare în timp real: Redarea se realizează printr-un mecanism de tip callback. La fiecare interval de timp, placa de sunet solicită un bloc de date (buffer). Engine-ul mixează matematic eșantioanele (samples) din toate track-urile active.
- Mixarea (Summing): Semnalele audio sunt reprezentate ca vectori NumPy de tip float32. Mixarea presupune adunarea vectorilor corespunzători fiecărui track, ponderată de volumul individual al acestora.
- Optimizare: Pentru a preveni blocarea interfeței grafice, procesarea audio rulează pe un fir de execuție (thread) separat, prioritar.

3. Managementul proiectelor și persistența datelor

Pentru a permite salvarea și reluarea lucrului, s-a implementat clasa ProjectManager.

- Formatul Fișierelor: Datele sunt serializate în format JSON, sub un format custom de fișier (“.pydaw”).

- Fișierul de proiect stochează:
 - Căile absolute către fișierele audio sursă.
 - Setările fiecărui track (Volum, Pan, Mute, Solo, Culoare).
 - Configurația efectelor: Tipul efectului și valorile parametrilor acestuia.
 - Informații globale (BPM, Master Volume).

4. Sistemul de Undo/Redo (Command Pattern)

Pentru a oferi o experiență de utilizare fluidă și sigură, s-a implementat Design Pattern-ul Command.

- Fiecare acțiune modificatoare de stare (ștergere track, modificare volum, adăugare efect) este încapsulată într-o clasă derivată din interfața Command. Aceste clase implementează obligatoriu două metode execute() și undo().
- Stiva de Istoric: Clasa UndoStack gestionează două stive (una pentru Undo, una pentru Redo). Când o comandă nouă este executată, stiva de Redo este golită. Sistemul are o limită configurabilă (ex: 50 de pași) pentru a optimiza memoria.

5. Sistemul de efecte

Aplicația include un Rack de efecte modular, construit folosind Strategy Pattern. Clasa abstractă AudioEffect definește interfața comună, permitând engine-ului audio să proceseze o listă dinamică de efecte fără a cunoaște detaliile interne ale fiecărui.

Algoritmi implementați:

- Distors: Implementează modificarea amplitudinii semnalului printr-o funcție de transfer neliniară (waveshaping). S-a utilizat tehnica de Soft Clipping bazată pe funcția arctangent, care oferă o saturare similară amplificatoarelor analogice cu lămpi. Formula matematică este:

$$y[n] = (2 / \pi) * \arctan(x[n] * \text{Drive})$$

unde "Drive" este o valoare controlată de utilizator; amplifică semnalul înainte de clipping (Pre-Gain), împingând valorile în zona neliniară a funcției. Rezultatul final este o combinație liniară între semnalul curat și cel procesat, folosind knob-ul de "Mix" pentru a păstra din dinamica originală (implicit 100%, nu se păstrează semnalul original). Există și un mecanism de protecție/optimizare, pentru a economisi resurse CPU în cazul în care efectul nu este perceptibil (dacă drive ≤ 0.001).

- Delay simplu: Utilizează un buffer circular (Ring Buffer) pentru a stoca eșantioanele trecute și a le reintroduce în mix cu o întârziere specificată și o atenuare (Feedback). Un vector NumPy este prealocat pentru a stoca istoricul recent al audio-ului. Se folosesc doi indecs care avansează circular:

```
read_ptr = (write_ptr - delay_samples) % buffer_length
```

Semnalul citit ("întârziat") este reintrodus în buffer, multiplicat cu un factor feedback (ex: 0.4), creând efectul de repetiții multiple care se atenuează în timp.

- Egalizator parametric (EQ 3-Band): Egalizatorul este implementat folosind o cascadă de filtre digitale Biquad (IIR - Infinite Impulse Response). Spre deosebire de o simplă filtrare FFT, filtrele Biquad sunt mai rapide și introduc mai puțină latență.

Tipuri de Filtre:

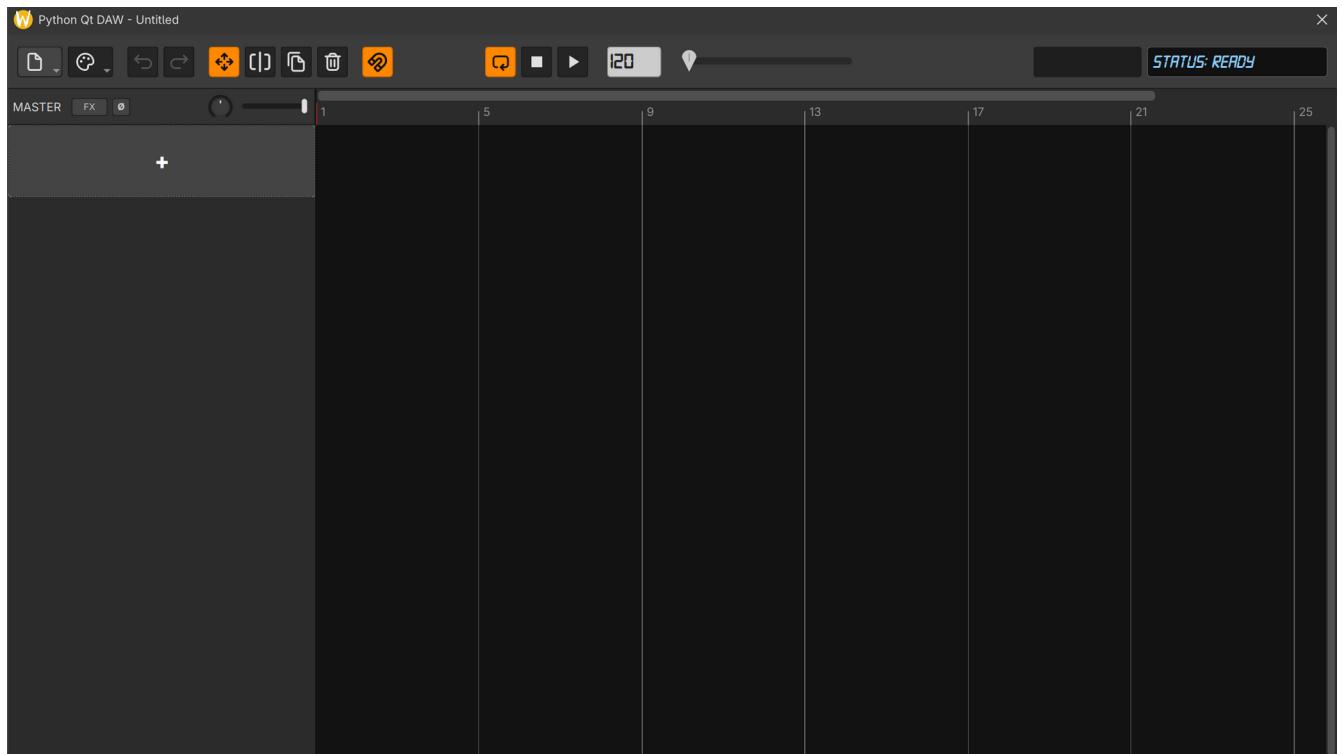
- Low Shelf: Controlează frecvențele joase (Bass). Atenuează sau amplifică tot ce este sub frecvența de tăiere (ex: 200Hz).
- Peaking (Bell): Controlează frecvențele medii (Mid).
- High Shelf: Controlează frecvențele înalte (Treble).

Frecvențele de tăiere pentru Low și High se setează prin butoane separate. Pentru procesare se utilizează funcția `scipy.signal.sosfilt` (Second-Order Sections), care este mai stabilă numeric decât formele directe I/II, evitând erorile de cuantizare la frecvențe joase.

IV. Utilizarea aplicației

Această secțiune descrie modul de interacțiune cu interfața grafică a aplicației și fluxul de lucru tipic pentru crearea unui proiect audio.

1. Interfața principală



Interfața la deschiderea programului

La rularea scriptului `main.py`, se deschide fereastra principală (Main Window) împărțită în trei zone funcționale distincte.

A. Bara de instrumente/Ribbon (sus). Conține, în ordine:

- “Project” – un dropdown din care se pot face diferite operații de manipulare a fisierelor de proiect (proiect nou, deschide proiectul, salvează, export).
- “Theme” – un dropdown din care se pot alege diferite teme, printre care și una pentru contrast ridicat.
- Undo și Redo – acestea apar dezactivate dacă nu există nimic în stiva lor.
- Tool-uri pentru manipularea secvențelor audio – “move”, “split”, “copy”, “delete”. Implicit este ales butonul “move”, care prin scurtături de la tastatură are la îndemână și capabilitatea celorlalte unele.
- “Snap” – un buton prin care clipurile audio pot fi inserate doar la început de măsură (sau $\frac{1}{4}$ din bătaie/minut). Implicit este activat.
- “Loop” – se redă de la început conținutul audio atunci când se termină. Implicit activat.
- “Stop” – oprește redarea și trimite indicatorul de redare la începutul proiectului.
- “Play/Pause” – pornește sau oprește redarea din punctul în care este setat indicatorul de redare.
- Selectorul de BPM (bătăi pe minut) – setează ritmul în care va parcurge tot conținutul proiectului. Se poate seta prin tastare directă, prin “click & drag” vertical sau prin alegerea unor valori arbitrară cu click-dreapta. Implicit este setat la 120 BPM.
- “Mini-Timeline” – un slider restrâns prin care se poate parcurge rapid întregul conținut. Este sincronizat cu redarea audio în timp real, cursorul deplasându-se proporțional cu durata totală a proiectului.
- Indicator de nivel (Stereo Meter) – un vizualizator în timp real pentru monitorizarea semnalului global de ieșire. Afisează două coloane independente pentru canalele stânga și dreapta, permitând verificarea echilibrului stereo al mixajului. Volumul este evidențiat de culori specifice în funcție de siguranța nivelului: verde – nivel sigur, galben – semnal puternic, roșu – apare clipping.
- Bară de status – un afișaj digital care oferă informații utile când se acționează diferite secțiuni sau butoane ale aplicației.

B. Panoul de Track-uri (stânga) este zona din care se vizualizează informații asupra pieselor și se controlează toate funcționalitățile care nu fac parte din sistemul de efecte. Deasupra tuturor track-urilor se află panoul “Master”, prin care se pot aplica efecte sau modifica volumul și pan-ul întregului proiect. Lista efectivă a track-urilor este implicit goală; apare doar butonul de adăugare. Fiecare track are următoarele funcționalități:

- “Color picker” – o bară colorată care personalizează culoarea clipurilor audio pentru diferențiere
- “FX” – buton care deschide panoul pentru efecte. Aceasta indică vizual prin culoare dacă există efecte aplicate track-ului respectiv, dar și numărul de efecte din rack.
- “Bypass” – dezactivează întreg rack-ul de efecte pentru a compara sunetul original cu cel procesat.
- “Mute” – oprește sunetul track-ului
- “Solo” – oprește sunetul tuturor celorlalte track-uri
- “Remove” – șterge întreg track-ul
- “Pan” – setează panoramificarea sunetului stereo, spre stânga sau dreapta.
- Fader de volum – setează volumul sunetului. Are integrat un vizualizator pentru nivelul volumului, similar cu cel din ribbon, dar pe un singur canal (mono). Implicit este setat la maxim.

Panoului “Master” îi lipsesc “color-picker-ul”, “Mute”, “Solo” și “Remove”. Acesta are și o dimensiune diferită de restul pentru integrarea vizuală cu alte elemente din Timeline.

Lățimea întregului panou se poate modifica prin drag cu mouse-ul de marginea dinspre Timeline. Are o dimensiune minimă și maximă arbitrară.

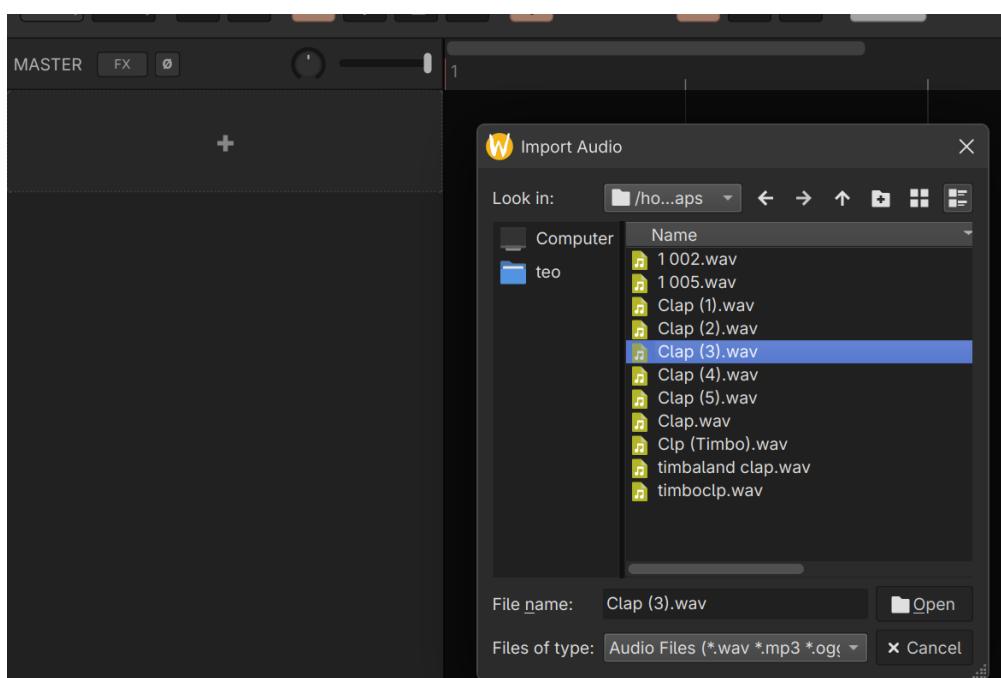
C. Timeline-ul (dreapta) este spațiul principal de lucru. Acesta este compus la rândul său din:

- Bară de scroll orizontală (sus) – parcurgerea în timp a proiectului.
- Bară de scroll verticală (dreapta) – parcurgerea în lista track-uri a proiectului.
- “Timeline Ruler” – o bandă orizontală situată deasupra zonei clipurilor, gradată în bătăi (beats). Pe suprafața acesteia se definește “Playhead-ul”, o bară roșie care indică momentul exact care este redat sau editat. Acesta are o prelungire mai subțire care trece vertical prin toate track-urile. La momentul play-ului, capul roșu începe parcurgerea, iar în urma lui rămâne o bară similară, dar albastră, care indică momentul de unde a pornit redarea, de unde s-a dat click.
- Zona clipurilor (Track Lanes) – fiecărui track îi corespunde o zonă orizontală (un “Lane”) în Timeline în dreptul panoului unde se pot prelucra clipurile. Prin clip se înțelege atât secvența audio implicită care apare la adăugarea track-ului, dar și părțile care reies prin prelucrarea cu uneltele disponibile. Clipurile nu pot fi mutate pe alte lane-uri. Efectele aplicate din dreptul panoului se aplică tuturor clipurilor. Vizual, un clip este un dreptunghi colorat, implicit albastru, cu o formă de undă reprezentativă în interior. La apăsarea unui clip culoarea acestuia se deschide, pentru a evidenția vizual faptul că este în “focus”.

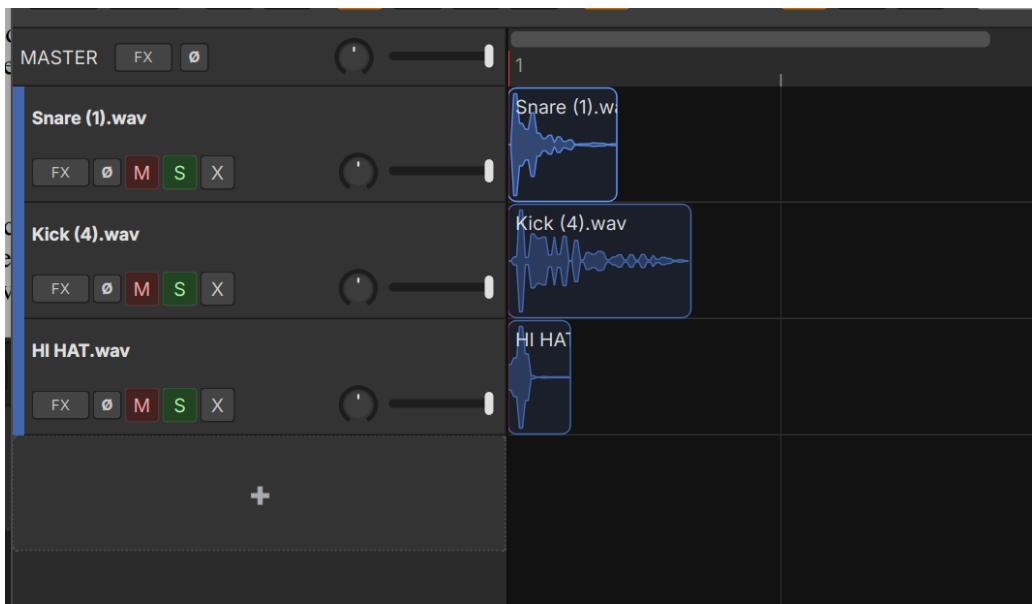
2. Workflow

A. Setările proiectului

Programul deschide implicit un proiect nou, care nu conține niciun fișier audio. Acestea se adaugă prin butonul mare cu “+” din zona din stânga. Se deschide un dialog specific sistemului de operare prin care se pot încărca fișiere audio diferite (.mp3, .wav, .opus etc).



Track-urile devin astfel populate și apar clipuri pentru fiecare dintre ele.



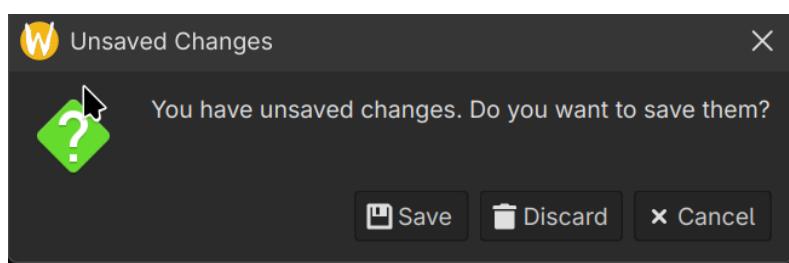
(Sunetele scurte, de genul tobelor s-ar putea să se vadă prea mici la crearea unui proiect nou. Se poate folosi zoom cu Ctrl+Scroll sau Ctrl+-/+=).

Proiectul nou se poate salva din dropdown-ul File → Save Project sau Save Project As, ori prin shortcut-ul Ctrl+S.

Orice modificare nesalvată a proiectului este evidențiată printr-un indicator (*) în dreptul numelui ferestrei. Aici se poate vedea și numele proiectului.



Orice acțiune de a părăsi un proiect nesalvat, precum deschiderea unui alt proiect, proiect nou sau închiderea programului, va atenționa utilizatorul în scopul salvării acestuia.

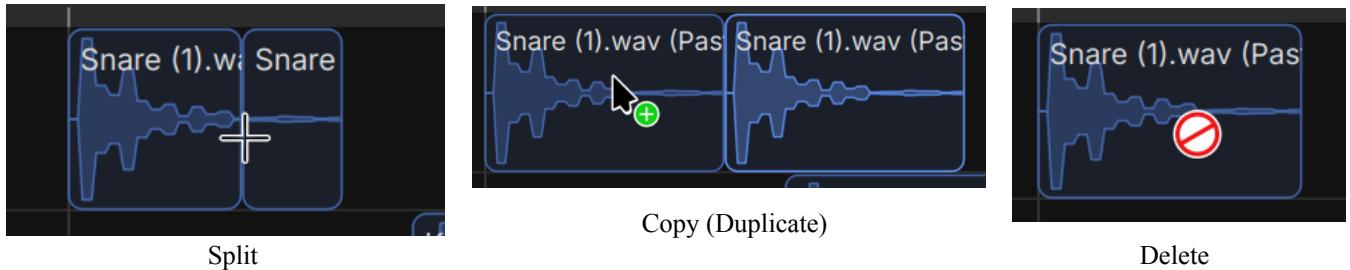


B. Prelucrarea clipurilor

Implicit, metoda de editare a clipurilor este “Move”. Aceasta este o unealtă puternică și poate fi folosită împreună cu shorcut-uri de la tastatură:

- Click pe clip, apoi click pe spațiu liber: copiază clipul selectat
- Shift+Click: taie clipul în punctul selectat
- Click-dreapta: șterge clipul

Existența restului de unelte poate fi redundantă în acest context, dar existența vizibilă a opțiunilor pentru utilizator poate fi importantă. Copierea este singura relativ unică; duplică clipul fix după terminarea lui (shortcut Ctrl+B). Fiecare prezintă cursori diferiți la activarea lor.



Lungimea fiecărui clip poate fi ajustată prin “handle-uri” care apar la începutul și sfârșitul lor, evidențiate printr-un cursor diferit.



C. Aplicarea de efecte

Panoul fiecărui Track prezintă un buton pentru efecte. Acesta deschide o fereastră nouă din care se pot adăuga efectele din listă. Fiecare efect are un buton de on/off, un buton de stergere și un “handle” prin care se poate schimba ordinea în rack.

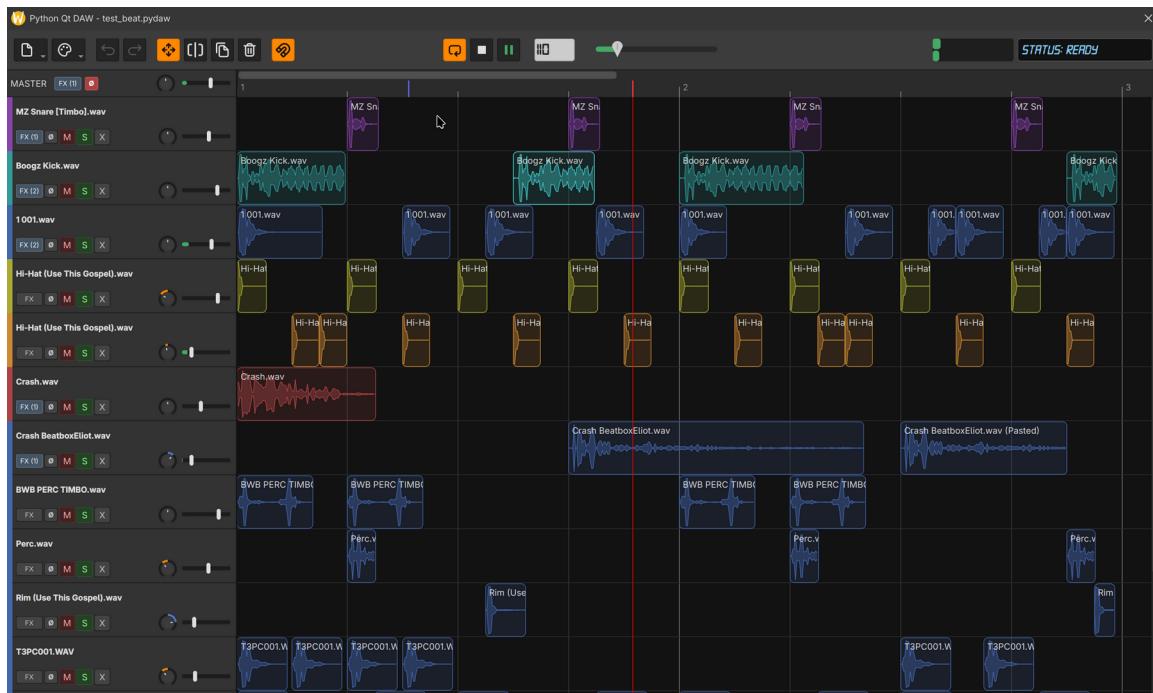


D. Alte detalii care îmbunătățesc workflow-ul

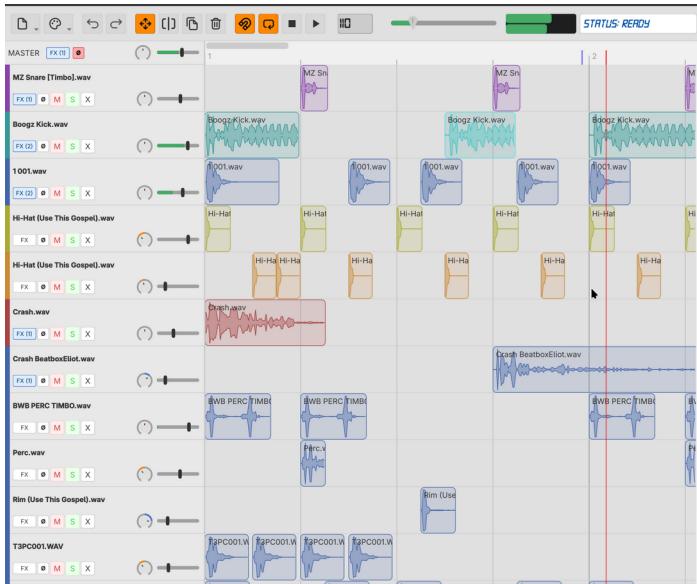
- La deschiderea unui proiect salvat, nivelul de zoom se raportează la lungimea totală a proiectului.
- Lungimea spațiului de lucru este calculat dinamic în funcție de poziția ultimului clip din timeline și de dorința utilizatorului de a da play mai mult în timp. Durata crește cu câte o măsură întreagă (4 bătăi), dând impresia de spațiu de lucru nelimitat.
- Utilizatorul poate trage de Playhead pentru a se deplasa mai precis în timp. Dacă Playhead-ul este apăsat și cursorul trece în stânga sau dreapta de o margine a spațiului de lucru se începe scroll-ul.

Rezumat al shortcut-urilor:

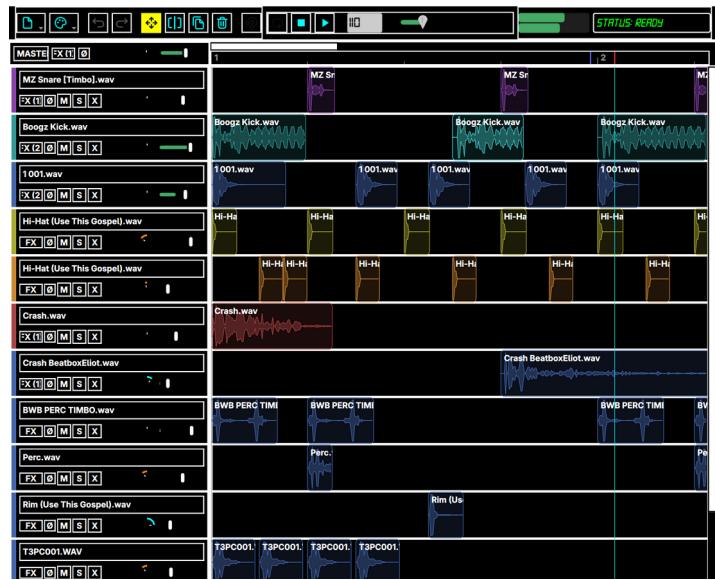
CTRL + S	Save
CTRL + SHIFT + S	Save As
CTRL + MOUSEWHEEL	Zoom In/Out (cursor)
CTRL + -/=	Zoom In/Out (playhead)
CTRL + B	Duplicate Clip
CTRL+Z	Undo
CTRL+Y / CTRL+SHIFT+S	Redo
SHIFT + LCLICK (clip)	Split Clip
RCLICK (clip)	Delete Clip
SPACE (global)	Play/Pause
MOUSEWHEEL	Scroll Vertical
ALT + MOUSEWHEEL	Scroll Horizontal



Exemplu – temă întunecată (implicit)



Exemplu – temă luminoasă



Exemplu – temă contrast ridicat

V. Dependențe și mediu de execuție

Pentru rularea și dezvoltarea aplicației au fost utilizate următoarele tehnologii:

- Limbaj de programare: Python 3.10.12 (versiune minimă recomandată 3.10)
- Framework GUI: PySide6 (v6.6.1) – randarea interfeței Qt6 și gestionarea evenimentelor
- Procesare numerică: NumPy – manipularea vectorilor audio și mixare
- Procesare semnal: SciPy – implementarea filtrelor IIR (egalizator)
- sounddevice – flux audio în timp real via PortAudio
- soundfile – citirea și scrierea fișierelor audio (WAV, MP3, FLAC)
- IDE utilizat – Visual Studio Code
- Sistem de Operare – dezvoltat și testat pe Linux (Arch + KDE), compatibil cu Windows/MacOS datorită naturii cross-platform a bibliotecilor utilizate.

Modalitatea de execuție:

Aplicația se livrează ca un pachet de surse Python, împreună cu asset-uri sub formă de fișiere .svg și font-uri, precum și fișiere .qss pentru tematică. Pentru execuție se recomandă utilizarea unui mediu virtual (venv).

- Se instalează dependențele:

```
pip install -r requirements.txt
```

- Se lansează aplicația:

```
python main.py
```

VI. Concluzii

Aplicația dezvoltată reprezintă un editor audio multitrack simplu, dar funcțional, capabil să gestioneze sesiuni relativ complexe de mixaj. Proiectul este ușor de extins datorită utilizării Design Pattern-urilor, astfel că eventuale extinderi devin posibile, precum o funcționalitate de înregistrare, integrarea efectelor standard în industrie (LV2, VST) sau implementarea unui motor audio mai performant și eficient cu ajutorul limbajelor de nivel scăzut (C++, Cython). Interfața cu utilizatorul este, de asemenea, un punct care ar putea fi îmbunătățit, cu opțiuni pentru vizibilitate (scalabile, dimensiunea textului) sau mai multe widget-uri personalizate pentru similaritate între sisteme.