Interactive techniques for level editing of sonification games

Qinxin Ren

Delft University of Technology

TU Delft

Interactive techniques for level editing of sonification games

by

Qinxin Ren

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on *[18th, December, 2025]*.

Advisor:	Rafa Bidarra
Project Duration:	Feb, 2025 – Dec, 2025
Faculty:	Faculty of Computer Science, Delft University of Technology

Composition of the thesis committee:

Chair:	Dr. R. Bidarra, Computer Graphics & Visualisation, TU Delft
Core Member 2:	Dr. C. Liem, Multimedia Computing, TU Delft
External Advisor:	Prof. R. Schaefer, Leiden University

TU Delft

Abstract

Gesture-based sonification games offer a promising medium for motor-skill rehabilitation by transforming therapeutic movements into interactive musical experiences. However, creating custom levels for such games remains inaccessible to therapists and researchers, who often lack the musical or technical expertise required to shape musical material into meaningful motor tasks. This thesis investigates how non-technical and non-musician users can be supported in generating music, authoring interaction structures, and validating playable levels for gesture-based sonification games.

To address this challenge, this work introduces an end-to-end authoring pipeline that combines symbolic music generation with an intuitive level-editing system. A hierarchical diffusion model is adapted to produce structured, multi-track musical material through high-level controls such as key, tempo, and song form. A web-based authoring interface then allows users to refine this material, simplify dense passages through a trigger–support note mechanism, and map notes to spatially and temporally aligned gesture targets. A target-configuration module provides synchronized previews and export functions that integrate directly with the *PIZZICATO* runtime, enabling real-time testing and performance logging.

A qualitative expert evaluation with nine therapists and researchers examined the usability, flexibility, and therapeutic potential of the system. Participants found the workflow accessible and intuitive, valued the direct manipulation of musical and spatial elements, and highlighted the potential of the tool to streamline content creation for motor-rehabilitation studies. The evaluation also surfaced conceptual limitations, including the need for broader musical genres beyond pop-derived structures, and the opportunity to incorporate clinically informed automation such as predefined motor-exercise patterns.

This thesis contributes (i) a novel, integrated workflow for non-technical authoring of gesture-based sonification levels, (ii) interface techniques that translate symbolic musical structure into spatial–temporal interaction tasks, and (iii) empirical insights into the needs of therapists and psychologists designing movement-based therapeutic content.

Contents

1

Introduction

1.1. Background

Motor skill rehabilitation is central to recovery for patients affected by neurological conditions such as stroke, traumatic brain injury, or neurodegenerative diseases. Traditional methods often employ repetitive exercises to restore movement and coordination. Although effective, these routines can become monotonous, risking reduced motivation and adherence over time.

Serious games offer an engaging alternative by transforming repetitive therapy into interactive challenges that encourage sustained practice. Among these, *sonification-based rehabilitation games* use sound as feedback for movement, providing immediate, intuitive cues that reinforce timing and coordination. In particular, *gesture-based sonification games* map discrete body movements to musical events in real time. Players interact with on-screen note targets by performing specific gestures—such as pinches or hand poses—detected by a standard webcam or similar sensors. Correctly timed actions are "sonified" as musical notes, coupling rhythm and movement to promote repetition, sustain motivation, and align therapeutic actions with clear temporal goals in an entertaining and engaging way.

1.2. Motivation

Although gesture-based sonification games have proven effective in both clinical and research settings, therapists and psychologists who wish to use such game mechanisms for their patients still face several practical challenges.

Limited content and flexibility

Current sonification games, such as **PIZZICATO**, offer only a small number of embedded songs—typically four—each with a short duration and simple melodic structure. This limited repertoire restricts therapeutic variety: practitioners lack access to music that matches the diverse needs and progress levels of their patients. In rehabilitation, however, a rich selection of levels and musical material is crucial to maintain motivation, adapt to different stages of motor recovery, and tailor difficulty to individual mobility goals.

Authoring complexity and dependence on specialists

Even when therapists or psychologists attempt to create new levels, they encounter two major barriers—musical and technical.

From the **musical side**, not all pieces are suitable for gesture-based sonification. The music must exhibit a clear pulse, separable layers, and salient note onsets to synchronize with gestures. Existing pre-recorded songs (e.g., MP3 files) not only introduce licensing and preprocessing burdens—such as tempo alignment, quantization, and segmentation—but also often fail to yield musical material that aligns well with sonification mechanics. As a result, practitioners must either possess compositional

expertise or have access to a large, license-free song library meeting these strict structural criteria.

From the **technical side**, once suitable music is identified, translating it into a playable, therapeutically meaningful level typically requires specialized knowledge and multiple external tools. Non-technical users lack intuitive ways to map musical structure onto interactive gesture targets, adjust timing and spatial layouts for different motor exercises, and export the resulting design into the game itself. Consequently, therapists and psychologists remain dependent on developers to implement their ideas, slowing experimentation and limiting clinical adaptability.

1.3. Research Question

Building upon the challenges outlined above, this study is guided by the following main research question:

> **How can tools and methods be designed and integrated to enable non-technical, non-musician users—such as therapists and researchers—to create and customize levels for gesture-based sonification games?**

This overarching question can be further divided into two complementary domains: the **music generation side**, which concerns how users obtain suitable musical material, and the **level design side**, which concerns how that material can be translated into playable, therapeutically meaningful levels.

Music material

To support therapists and psychologists in obtaining appropriate musical material, the research investigates the following questions:

1. What musical characteristics make a piece suitable for use in gesture-based sonification games (e.g., clarity of pulse, separable layers, salient onsets)?
2. How can non-musician and non-technical users acquire or generate such music material without requiring compositional or production expertise?

Level design

Once suitable music is available, the next challenge lies in enabling users to design levels that align with patients' therapeutic goals. The research therefore explores:

1. Which interface features and representations can help non-technical, non-musician users translate musical structures into interactive levels in an intuitive way?
2. What degree of flexibility and control is needed for users to adjust parameters such as difficulty, timing tolerance, and spatial layout to fit individual rehabilitation needs?
3. How can users test and validate their designed levels within the game environment to ensure playability and therapeutic appropriateness?

1.4. Objectives

To address the research questions, this thesis pursues three main objectives, corresponding to the two technical components of the system and their subsequent evaluation.

O1 **Music generation.** Develop an intuitive method that enables therapists and psychologists to generate suitable musical material for gesture-based sonification games without requiring musical or technical expertise. The generation process should minimize the use of specialized terminology, reduce cognitive and operational load, and provide users with controllable yet accessible means to obtain task-appropriate music.

O2 **Level editor.** Design and implement an interactive editor that allows non-technical and non-musical users to translate generated music into playable game levels. The editor should support the assignment of note events to spatial layouts and specific gestures in a clear and intuitive manner, requiring no programming or compositional skills.

O3 **Evaluation with domain experts.** Assess the system's usability, flexibility, and integration within therapeutic and research workflows through expert walkthroughs and usability testing with thera-

pists and researchers. Gather and analyze qualitative feedback to guide iterative refinement of both the music generation and level editing components.

1.5. Structure of the Thesis

The remainder of this thesis is organized as follows:

Chapter 2: Related Work. Surveys gesture-based sonification for rehabilitation and learning, authoring approaches in serious games, and controllable symbolic music generation to situate the thesis.

Chapter 3: Design Constrains & Requirements. Defines user needs (therapists, researchers), derives design goals, and specifies high-level requirements for music creation and level authoring. Outlines the end-to-end pipeline architecture linking music generation, level creation, and runtime integration.

Chapter 4: Music Generation. Describes the approach to obtaining suitable musical material, including a brief analysis of explored alternatives and the rationale for the chosen strategy based on the failed attempts, with emphasis on criteria important for gesture-timed interaction.

Chapter 5: Level Authoring & Integration. Presents the principles and workflow for turning musical material into playable levels and explains how authored content is prepared for execution in a gesture-based sonification game.

Chapter 6: Implementation. Summarizes the implemented features and interface characteristics that realize the authoring pipeline in practice and support practical use by non-technical and non-musician authors.

Chapter 7: Evaluation. Details the expert evaluation setup and outcomes, focusing on usability, flexibility, and integration for clinical and research contexts.

Chapter 8: Discussion & Conclusion. Interprets the findings, reflects on limitations, and identifies directions for future work on accessible authoring for gesture-based sonification games.

1.6. Acknowledgement

This chapter was entirely drafted and written by the author. ChatGPT was used only for phrasing and minor language refinement. All ideas, analysis, and arguments are the author's own. Prompts were like: help me to check the grammar and improve the phrasing.

2

Related work

This chapter reviews the literature underlying this thesis across two complementary strands: (i) sonification for motor-skill learning and rehabilitation, and (ii) gesture-based sonification games, including musical material, authoring tools, and remaining challenges. Together, these strands motivate the need for accessible, end-to-end authoring workflows that let non-technical and non-musician users design gesture-based sonification content.

2.1. Sonification in Motor Skill Learning

Sonification transforms movement into sound, offering an alternative sensory channel that supports perception, timing, and motivation in motor-skill learning. This section examines how auditory feedback contributes to movement control, how musical structures enhance engagement, and which acoustic and perceptual properties make sound effective for learning tasks. Together, these studies establish the foundation upon which interactive and gamified sonification systems are later built.

Auditory Feedback as a Driver of Motor Learning

Sonification—the transformation of movement into sound—has been widely studied as a mechanism to enhance motor skill learning. By providing real-time auditory feedback, it makes the temporal and spatial characteristics of motion perceptually explicit, enabling learners to detect errors, adjust timing, and refine coordination with minimal cognitive load. This process strengthens the perception–action coupling fundamental to motor learning, as movements immediately yield sensory consequences that reinforce accurate performance [5, 4]. Empirical studies demonstrate that well-designed auditory mappings improve timing precision and retention of complex motor tasks by guiding attention toward relevant kinematic features and facilitating adaptive correction [7].

Effective implementations emphasize temporal precision, ecological relevance, and perceptual clarity, ensuring that auditory changes correspond directly to meaningful movement parameters such as speed, amplitude, or trajectory. Such mappings enable users to form stable internal representations of motion dynamics, accelerating motor learning through multimodal integration. Furthermore, studies indicate that participants find auditory feedback inherently engaging, reporting improved focus, motivation, and confidence when their movements elicit clear and consistent auditory responses [2, 12]. While such feedback improves movement control and perception, its motivational impact becomes most pronounced when embedded within structured, rhythmically coherent sounds such as music.

The Role of Music

Music transforms sonification from a purely informational feedback signal into an emotionally and motivationally engaging experience. Its rhythmic and harmonic structure provides a temporal scaffold that guides movement while sustaining user motivation. Rhythmic auditory stimulation (RAS) and music-supported therapy have long demonstrated benefits for motor rehabilitation, helping participants synchronize movements and maintain consistent timing [14].

Beyond rhythmic entrainment, music activates reward and emotion networks in the brain, enhancing engagement and adherence to repetitive tasks [21]. Melodic and harmonic cues can also facilitate learning through auditory prediction: when a listener anticipates the next event, movement timing becomes more precise [5]. This predictive link between sound and action is particularly valuable in rehabilitation, where continuous repetition is required for progress.

Overall, the role of music in sonification extends beyond aesthetics. It transforms feedback into a multisensory motivational system that supports timing, attention, and long-term engagement—key factors for sustaining motor learning through interactive and gamified systems.

Design Considerations for Effective Sonification

Building upon these perceptual and motivational foundations, researchers have proposed several design principles to ensure that sonification systems effectively support motor learning. Gross-Vogt et al. [9] identify four essential criteria for effective sonification design: (1) **perceptibility**—sounds should be easily distinguished; (2) **natural mapping**—auditory parameters should vary intuitively with gesture dynamics; (3) **semantic consistency**—feedback must align with the user's task and expectation; and (4) **contextual fit**—sound characteristics should suit the target environment. Within motor learning, Dyer et al. [6] stress the need for low-latency, onset-precise sounds that highlight key motion features. Frid and Bresin [8] show that transparent mapping between gesture and sound promotes user understanding, while Kantan et al. [10] emphasize ecologically valid sound design—using familiar timbres and textures—to reduce cognitive load.

Practical use of sonification in rehibilitation

These mechanisms have been successfully applied in rehabilitation, where motor skill learning is used to restore movement after injury or neurological impairment. Real-time auditory feedback helps patients perceive errors, maintain rhythm, and sustain motivation during repetitive exercises. Studies in upper-limb rehabilitation demonstrate that musical sonification improves timing precision and user preference compared to silence or non-musical feedback [12]. Similarly, reviews of gait and balance sonification show that concurrent and meaningful auditory feedback consistently produces positive outcomes in motor performance and adherence [20].

2.2. Gesture-Based Sonification Games

Building on the perceptual and musical foundations discussed earlier, gesture-based sonification games translate physical gestures into real-time auditory feedback within structured, game-like environments. By combining movement, sound, and challenge, such systems make motor practice more engaging and measurable while preserving the therapeutic benefits of sonification.

Gamification in Gesture-Based Sonification

Gamification introduces elements of play—such as goals, scoring, and progression—into non-game contexts to enhance motivation and persistence. In motor-skill rehabilitation, these mechanics transform repetitive exercises into structured, rewarding activities that sustain engagement. Studies show that gamified interventions improve adherence and functional outcomes by increasing enjoyment and sense of control: Alfieri et al.,[1] reported greater motivation and compliance in gamified musculoskeletal therapy, while Pimentel-Ponce et al.,[13] observed faster recovery and higher satisfaction. Multisensory, feedback-rich environments further strengthen agency and immersion—key factors in motor learning [16]. Within this framework, sonification provides real-time auditory feedback that naturally fits game dynamics: correct movements trigger consonant or rewarding sounds, while errors yield silence or dissonance, forming a self-reinforcing feedback loop. This integration makes training both informative and intrinsically engaging, turning the acquisition of motor precision into an active, exploratory process rather than passive repetition.

Empirical studies highlight the motivational and perceptual advantages of this integration. Frid and Bresin [8] showed that transforming children's gestures into sound stimulated playfulness and supported motor coordination through intuitive mappings. Schoeller et al.,[15] reported that adults engaging in gesture-based sonification experienced heightened bodily awareness and intentional control, while Volta et al.,(2019) observed similar motivational gains in educational motion-based games. A represen-

tative implementation is the *PIZZICATO* system by Starkov et al.,[17], which employs webcam-based hand tracking to let players “pinch” on-screen targets in rhythm. Correctly timed gestures trigger immediate auditory feedback, reinforcing precision through sound. User studies reported high accessibility, enjoyment, and a strong sense of agency, demonstrating how gamified sonification effectively merges rehabilitation, learning, and play.

Design and Authoring Process and Challenges

Level design in existing gesture-based sonification games remains highly constrained. While some commercial rhythm games such as Guitar Hero or Osu! provide level editors, they typically demand substantial musical and technical expertise, limiting accessibility for therapists or educators. In research contexts, most sonification systems are developed as isolated prototypes, focusing either on sound mapping or gesture interaction rather than an integrated authoring workflow. As a result, there is still no complete pipeline that connects the generation of suitable musical material, its mapping to gesture-based sonification, and its implementation as a playable game environment tested in rehabilitation settings.

Several studies have explored generation and mapping methods for sonification, yet these efforts remain fragmented. For instance, Lopes and Yannakakis [11] and Cardinale et al.,[3] investigated procedural and AI-driven sonification frameworks for interactive applications, but these systems were not applied in clinical or motor-learning contexts. Kantan [10] emphasized the absence of standardized authoring tools for real-time sonification, noting that most implementations rely on ad hoc programming. Similarly, Vallejo-Pinto et al.,[18] integrated configurable sound feedback into the e-Adventure game authoring platform, yet the tool was designed for accessibility research rather than embodied rehabilitation. Although middleware such as Audiokinetic Wwise supports sophisticated audio integration for games, it requires expert knowledge of sound design and scripting. Consequently, gesture-based sonification games in rehabilitation still rely heavily on predefined content and expert development, with few tools allowing non-technical users to create or adapt levels according to specific therapeutic needs.

Developing an accessible authoring system is therefore essential to bridge the gap between research prototypes and real-world therapeutic use. Therapists, researchers, and educators often lack programming or music-production expertise but still require the ability to tailor sonification games to individual users’ needs—adjusting difficulty, pacing, gesture mapping, or auditory feedback according to patient ability or learning objectives. Without such customization, existing systems remain static and poorly suited to long-term rehabilitation, where adaptability and variation are key to maintaining motivation and tracking progress. A unified, low-threshold authoring environment would allow non-technical users to import or generate suitable musical material, configure gesture–sound mappings, and test levels interactively within a single workflow. This accessibility would not only expand the practical use of sonification games in therapy and education but also support reproducibility and systematic evaluation in motor-learning research.

2.3. Ackownledgement

During the preparation of this literature review, I used the OpenAI GPT-5 language model to assist in identifying relevant academic sources and refining the phrasing of my writing. The tool was employed to locate peer-reviewed publications, summarize key findings for efficiency, and improve the clarity and coherence of the text. All conceptual framing, critical analysis, synthesis of literature, and final interpretations presented in this thesis are entirely my own. The language model served only as a research and writing aid, comparable to a digital assistant for language editing and information retrieval.

3

Design Constraints and Requirements

3.1. Objective

This chapter establishes the design objectives that guide the development of a framework for authoring gesture-based sonification games. The primary aim is to determine how non-technical and non-musician users—such as therapists and psychologists—can be supported in composing and adapting such games for diverse therapeutic conditions.

Specifically, the chapter pursues three objectives: (1) to identify the user profiles and define their needs; (2) to formulate conceptual criteria that ensure usability, flexibility, and extensibility, and to discuss their instantiation in the context of **PIZZICATO**; and (3) to define the workflow architecture that connects music generation, level design, and interactive testing.

3.2. Intended users and core use cases

The intended users of this system are therapists and psychologists working with patients who experience mobility impairments or cognitive disorders, or who are conducting research on how music and games influence cognitive and motor behavior. Gesture-based sonification provides an engaging and effective medium for both therapeutic intervention and experimental study in these contexts.

However, these users often lack professional expertise in music composition or programming, while still requiring the ability to customize game levels to suit different therapeutic goals and patient needs. The proposed system is therefore designed to bridge this gap by enabling non-technical users to design and adapt gesture-based sonification games through an intuitive interface.

3.3. Design Criteria and Conceptual Goals

This section outlines the principal criteria that guided the design of the authoring framework. These criteria articulate the conceptual and methodological goals that ensure the framework meaningfully supports the study of gesture-based sonification for rehabilitation and research purposes. They address the usability for non-technical and non-musician users, the flexibility of the design process across therapeutic conditions, and the extensibility of the framework for future studies and integrations.

3.3.1. Usability

Usability in this study concerns how non-technical and non-musician users can intuitively engage with the creative process of designing gesture-based sonification games. The inquiry focuses on supporting therapists and psychologists—who often lack experience in music production or programming—through three central activities: (1) obtaining musically coherent material suitable for therapeutic use, (2) translating that material into a structured and playable game level, and (3) evaluating the designed level within a real game context.

The challenge lies in designing interaction principles that make these creative processes accessible

without relying on technical expertise. Instead of specialized musical or programming terminology, the workflow should communicate through visual and auditory cues that are immediately comprehensible. The objective is to identify how users can be guided by intuitive, perceptual feedback—adjusting rhythmic density, aligning gestures with sound events, or refining timing relationships—so that the overall process remains engaging and cognitively manageable.

3.3.2. Flexibility

Beyond ensuring usability, this study also considers flexibility as a core design criterion. Flexibility refers to the capacity of the authoring process to accommodate diverse therapeutic contexts and patient conditions. In rehabilitation practice, individuals exhibit varying degrees of motor impairment and cognitive ability; consequently, even a single musical piece may need to be adapted into multiple levels of difficulty.

The design inquiry therefore focuses on how musical and interaction parameters can be adjusted to match these differing needs. This involves exploring ways to modulate task complexity—such as altering the density or timing of interaction targets—while preserving the musical integrity and rhythmic structure of the original composition.

3.3.3. Extensibility

Extensibility concerns the capacity of the design framework to connect authoring and evaluation within a unified research process. Once a level can be created and adapted to different therapeutic conditions, it becomes essential to investigate how that level functions in an actual gesture-based sonification context. This criterion emphasizes the need for seamless integration between the authoring environment and an existing interactive platform—such as **PIZZICATO**—that enables real-time testing and observation of player performance. Through such integration, the designed material can be examined not only for playability but also for its therapeutic and experiential effects.

3.3.4. PIZZICATO-specific requirements

PIZZICATO provides a concrete instantiation of gesture-based sonification for therapeutic and research contexts. Within this general framing, *PIZZICATO* operationalizes sonification through pinch-based interaction. Each on-screen note is executed via a pinch gesture (thumb with other fingers); successful pinches produce sound events that accumulate to render the unfolding song. When timing is imperfect, actions are unable to be sonified, which provides informative auditory cues that support learning through contrast.

Building on this interaction, the gameplay structure follows a **layered loop**. Levels comprise multiple musical layers—beat, chords, melody—performed one at a time. When the active layer is completed to criterion, that track is mixed into the background and begins looping as accompaniment while the next layer becomes active. Authors can specify the completion thresholds of pinch accuracy for reaching the next layer to adjust the level difficulty.

3.4. Architecture Overview

Building upon the criteria discussed above, the overall architecture follows a linear pipeline that converts musical material into playable levels and executes them in a real game - *PIZZICATO*.

Figure 3.1 illustrates the end-to-end workflow that links music generation, editing, and therapeutic game deployment in a linear pipeline. The process begins with **Music Generation**, where users specify basic parameters (e.g., tempo, global form, and length) to produce an annotated, multi-layer symbolic piece. Concretely, the output consists of (i) a multi-track MIDI file and (ii) a lightweight sidecar text file that records the song's structural specification (e.g., key, section labels such as verse/chorus, and the duration of each section).

The generated music is then opened in the **Level Authoring** module, which consists of two pages: **Note Editing** and **Target Configuration**. In the Note Editing page, users refine note content and rhythmic density—for example, by grouping several melodic notes into a single on-screen target—and can optionally create multiple difficulty versions by reopening and modifying the saved annotated music. The refined material is then passed to the Target Configuration page, where users assign fingers and

spatial positions to each target with a live preview aligned to the in-game layout. The annotated level can be saved and reloaded in the interface to create additional versions or further adjustments.

Finally, the configured level is exported and loaded into *PIZZICATO* for runtime execution. Each uploaded level includes multi playable layers (e.g., beat, chords, melody), and each layer has multiple targets for the player to pinch and sonify. Together, this pipeline enables a seamless transition from musical idea to a playable, measurable rehabilitation experience.

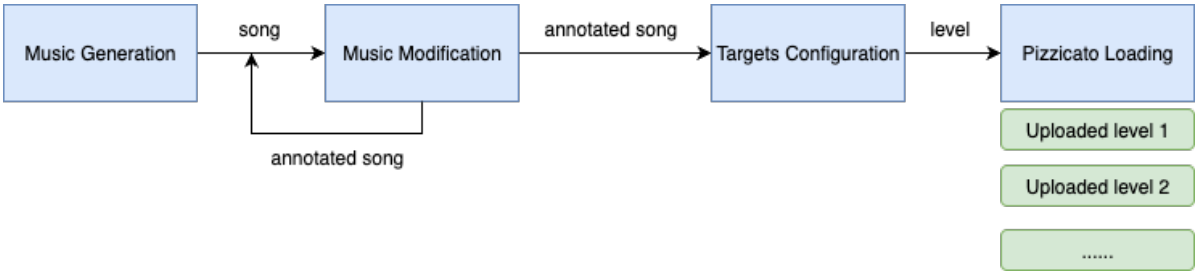

Figure 3.1: System architecture and data flow from music generation to gameplay in *PIZZICATO*.

3.5. Acknowledgement

This chapter was entirely drafted and written by the author. ChatGPT was used only for phrasing and minor language refinement. All ideas, analysis, and arguments are the author's own. Prompts were like: help me to check the grammar and improve the phrasing.

4

Music Generation

4.1. Objective

This chapter addresses the first stage of the overall pipeline—music generation—and specifically aims to: (1) identify the musical characteristics required for integration within the gesture-based sonification gameplay mechanism; and (2) investigate how music with these characteristics can be made accessible to therapists or researchers who may not possess prior experience in music composition or technique.

4.2. Design Rationale

4.2.1. Musical Requirements

In gesture-based sonification games, musical sounds are not merely background elements but are directly triggered by the player's movements. Each action produces a sonic event which is separated from a complete music piece that provides immediate feedback, allowing players to perceive and adjust their motor performance in real time. Typically, these sounds correspond to individual musical notes, which together form a coherent musical sequence that is both rhythmic and engaging.

From this gameplay mechanism, we identify that suitable musical material must exhibit **clear, precise, and separable note events**, enabling each note to be individually sonified and accurately triggered through gesture. Furthermore, the musical layers should remain sufficiently distinct to prevent excessive overlap, ensuring that rhythmic and harmonic information can be clearly perceived during movement.

Early explorations focused on **automatic note extraction from audio recordings**. Using tools such as Spleeter and Librosa, we attempted to decompose existing MP3 songs into melodic, harmonic, and percussive components. However, this approach proved unsuitable for therapeutic sonification: even after source separation, the resulting onsets were often blurred, frequency overlapped, and transcribed MIDI data lacked the temporal precision required for gesture-based interaction. Consequently, the extracted material failed to provide the *clear and isolated note events* necessary to support precise motor gestures during gameplay.

These limitations motivated a shift from extracting notes from audio files to directly using **MIDI-format songs**, which explicitly encode pitch, onset, and duration for every note. To address copyright and flexibility concerns, we further explored **algorithmic MIDI generation**, allowing users to obtain musically coherent material by adjusting a small set of high-level parameters rather than composing manually.

4.2.2. Music Generation Method

Given the limitations of using pre-existing audio material, this research explores how algorithmic music generation can provide both flexibility and accessibility for non-musician users. The goal is to enable users—such as therapists and psychologists—to obtain musically coherent material without requiring compositional or technical expertise.

To avoid overwhelming users with complex musical terminology, the generation process should be designed around a minimal set of intuitive, high-level parameters. These parameters correspond to perceptually meaningful musical dimensions—such as *tone*, *tempo*, and *structural length*—rather than detailed theoretical constructs. In addition, explanatory visual or auditory feedback should accompany each step to help users understand the effect of their choices and maintain an overall sense of the musical form they are generating.

Since therapists and researchers may have limited or no musical background, the workflow emphasizes clarity and guidance. When the use of musical terminology is unavoidable, it is supported by brief textual explanations or visual cues to illustrate the concept. This design principle enables users to progressively refine the generated music through a few simple interactions, without requiring prior expertise.

4.3. Approach

The generation method follows a straightforward procedure: users configure the basic musical parameters, trigger the generation process, and download the resulting output, which can later be imported into the level editor for integration and testing.

4.3.1. Music Generation Algorithm

Building upon the design rationale outlined in the previous section, this study adopts a **hierarchical symbolic generation** approach that constructs music progressively—from high-level form and phrase structure down to individual note realizations—ensuring both global coherence and local rhythmic clarity.

The **Whole-Song Hierarchical Generation** framework [19] represents the first attempt to model complete pop songs—including both melody and piano accompaniment—under an explicit realization of compositional hierarchy. It defines a four-level hierarchy of symbolic music representations, where each level captures contextual dependencies at a distinct musical scope, ranging from high-level song form and phrase structure to low-level notes, chords, and local rhythmic patterns. A cascaded diffusion model is trained to model these hierarchical levels, with each level conditioned on its upper layer. This design allows the system to maintain long-range structural coherence while generating musically consistent local details, enabling the creation of full-length, multi-track symbolic compositions with clear form, phrasing, and harmonic continuity.

Through preliminary testing, we observed that the *lead sheet* and *reduced lead sheet* levels of the hierarchy best satisfy the requirements for sonification gameplay. In contrast, the final accompaniment level produces overly dense arrangements containing excessive note events, which reduce clarity and precision during gesture-based interaction. Consequently, the algorithm was modified to render outputs only up to the lead sheet stage. During level creation, users can choose between the reduced and standard lead sheet versions according to the desired level of rhythmic and harmonic complexity.

4.3.2. Integration with Sonification Design

After the generation process, the user obtains a complete musical piece that serves as the foundation for subsequent level authoring. The generated MIDI file can be directly downloaded, played locally for inspection, or imported into the level editor for visualization and interactive preview. Within the editor, the structural markers embedded in the file (e.g., *intro*, *verse*, *chorus*, *bridge*) are automatically recognized and displayed in different colors, allowing users to organize and manipulate musical sections with minimal effort.

This integration ensures that algorithmic music generation is not an isolated procedure but an integral part of the end-to-end workflow—from composition and gesture mapping to interactive sonification and testing. In this way, the generated music functions as both a compositional and therapeutic scaffold, bridging algorithmic creativity with embodied interaction in the sonification game - **PIZZICATO** environment.

4.4. Discussion

This chapter examined the ideal musical characteristics required for gesture-based sonification games and explored how non-technical and non-musician users can obtain suitable musical material through an accessible generation process. The hierarchical music generation method proved highly effective in meeting these requirements, with several modifications incorporated into the generation procedure to better align with the specific needs of sonification-based interaction. In addition, the discussion highlighted the importance of guidance to help users understand and navigate both the generative workflow and the integration process necessary for evaluating the produced music.

The detailed implementation settings and interface workflow for preparing and evaluating the generated material are presented in Chapter 6.

4.5. Acknowledgement

This chapter was entirely drafted and written by the author. ChatGPT was used only for phrasing and minor language refinement. All ideas, analysis, and arguments are the author's own. Prompts were like: help me to check the grammar and improve the phrasing.

5

Level Design

5.1. Objective

After obtaining the desired generated song, this phase investigates how the generated musical material can be transformed into *playable and therapeutically meaningful* interaction tasks within a gesture-based sonification game—**PIZZICATO** in this case.

The primary objective of this stage is threefold: first, to identify the essential components required for composing a gesture-based sonification level; second, to establish design principles that enable non-technical and non-musician users—such as therapists or researchers—to author levels in an intuitive and accessible way; and third, to determine how users can preview and transform the generated levels into playable experiences for their patients.

5.2. Design Rationale

5.2.1. Key Steps of Level Composition

In the gesture-based sonification gameplay mechanism, each musical note is translated into a *playable target* displayed on the game canvas for the player to interact with and subsequently sonify. This mapping process defines four fundamental factors in level composition: (1) **when** each target should appear (temporal placement), (2) **where** it should be located on the screen (spatial placement), and (3) **which gesture** should be assigned to trigger it (gesture mapping). (4) **what sound** is being triggered based on the on-time gesture (auditory mapping)

Accordingly, these four factors form the essential parameters of the level authoring procedure. They represent the minimal yet sufficient set of components that users must be able to configure when designing new levels, ensuring that each level maintains both rhythmic and motor coherence within the sonification framework.

5.2.2. Accessibility Design Principles

To realize the four key steps of level composition, it is essential to consider how these elements can be made accessible to non-technical and non-musician users. Rather than overwhelming users with musical or programming terminology, the system must communicate through intuitive and perceptually meaningful interactions. The authoring process therefore emphasizes *direct manipulation* and *immediate feedback*, allowing users to create and adjust levels in a visually guided manner without technical intervention.

For spatial and gestural editing, the interaction should also prioritize perceptual intuition over numerical precision. Users should be able to interact directly with visual elements corresponding to musical or gestural events, rather than through parameter lists or text fields. This ensures that the process of placing and assigning targets follows the same perceptual logic as gameplay itself.

Finally, to foster understanding and iteration, users must be able to **preview** their authored levels before

deployment. A real-time visualization of timing and sequence allows them to perceive how targets will unfold during gameplay, providing immediate insight into rhythmic balance and overall playability.

5.2.3. Flexibility Design Principles

Since patients' physical and cognitive conditions may vary widely, the level authoring system must support flexible adaptation of difficulty and complexity without altering the underlying musical and rhythmic integrity. Flexibility in this context refers to the system's ability to accommodate different therapeutic goals with various motor capacities while maintaining coherent and engaging musical structures.

The system should also provide users with the means to refine or simplify the generated material at varying levels of granularity. Adjustments to timing, phrasing, or density enable designers to align the musical interaction with the intended therapeutic outcomes, ensuring that levels remain both musically expressive and physically achievable. This flexibility supports a personalized approach to rehabilitation, where musical structure and motor challenge can be co-designed to meet individual needs.

5.2.4. Extensibility Design Principles

Beyond authoring and customization, the design must also ensure that created levels can be seamlessly integrated into the actual gameplay environment for testing and evaluation. Extensibility therefore concerns the system's capacity to connect the authoring interface with the runtime environment in a transparent and maintainable way.

Conceptually, this principle ensures that the authoring tool is not an isolated editor but a component of a continuous workflow—from music generation, through level design, to therapeutic application. Each stage should exchange information in a consistent, interpretable format so that users can directly experience and assess their designs within the same framework used for patient testing.

5.3. Approach

To implement the design principles outlined above, this study adopts an iterative design approach that treats level authoring as a translation of symbolic musical material into interactive motion tasks. The workflow connects algorithmic music generation with embodied interaction in **PIZZICATO**, where each on-screen target corresponds to a pinch gesture and produces immediate auditory feedback upon correct timing.

The proposed authoring process consists of four main stages: mapping and simplification, gesture assignment and spatial layout, preview and iteration, and exportation with runtime integration.

5.3.1. Note Mapping and Simplification

The first stage focuses on adapting the generated musical material into a form suitable for gameplay. To balance musical richness with physical feasibility, the system introduces a **trigger–support note mechanism**. Each note in the generated song is categorized as either a *trigger note*—which becomes a playable target—or a *support note*, which is musically linked to its corresponding trigger to form a short melodic phrase.

When a trigger is successfully activated during gameplay, all associated support notes are sonified automatically, producing the full musical phrase. If the trigger is missed, the phrase is skipped entirely. This approach maintains rhythmic continuity and melodic coherence while reducing the number of required gestures, thereby preventing excessive motor load for patients with limited mobility.

The editor also allows users to refine the generated musical material. They can trim less relevant sections (e.g., intros, bridges, or outros) to adjust the overall song duration, or modify individual notes by editing their pitch, duration, or onset time. These editing capabilities ensure that the resulting level stays firmly grounded in the musical structure of the desired song.

5.3.2. Gesture Assignment and Spatial Layout

Once the musical events are mapped and simplified, users proceed to define the spatial and gestural aspects of the interaction. Each trigger note is assigned to a specific gesture, typically corresponding to one of the finger–thumb pinch combinations recognized in **PIZZICATO**.

The editor provides a **visual authoring canvas** that mirrors the game's interaction space. Within this interface, users can directly place and arrange targets at their desired screen positions using a drag-and-drop operation, ensuring that spatial configuration remains intuitive. The system automatically records the temporal, spatial, and gestural parameters of each target.

Real-time visual alignment indicators help verify that targets appear at musically synchronized moments and within ergonomic reach zones. This stage therefore bridges symbolic musical structure with motor behavior design, translating abstract rhythm into spatialized, performable actions.

5.3.3. Preview and Iteration

To facilitate rapid testing and understanding, the system provides a built-in **preview time window** that replicates the temporal behavior of the actual gameplay. In this mode, users can visualize both the current and upcoming targets as in **PIZZICATO**.

The preview window corresponds directly to the in-game timing model: the left side represents the present playback position, while the right side extends forward over a defined preview duration. All notes whose onset times fall within this window are displayed on the canvas, their visual brightness gradually decreasing in proportion to temporal distance—allowing users to perceive upcoming targets in a continuous, time-based manner. In addition, users can isolate any musical section to display all targets within that section, enabling a focused examination of rhythmic or spatial patterns before testing.

This preview functionality enables users to evaluate rhythmic synchronization, gesture assignments, and spatial arrangement in real time without leaving the authoring environment. Such immediate, context-aware feedback supports iterative refinement and encourages an exploratory design process, even for users without technical or musical expertise.

5.3.4. Exportation and Runtime Integration

The final stage focuses on transferring authored content to the runtime environment for testing and deployment. Once satisfied with the preview, users can **export** their level directly into **PIZZICATO**. The export process automatically packages the level's data—including temporal, spatial, and gestural information—into a format readable by the game engine.

In the runtime, levels follow a **layered progression structure**: players begin with basic chord layer and unlock additional melody layer as they achieve predefined accuracy thresholds. This progressive system sustains motivation and allows adaptive difficulty, aligning the musical experience with the patient's evolving motor control.

Through this seamless integration between the editor and runtime, users can quickly test levels with real participants, observe performance metrics, and re-import feedback into the authoring environment for further refinement. This end-to-end loop—generation, authoring, preview, and testing—ensures that musical, interactive, and therapeutic objectives remain continuously aligned.

5.4. Discussion

Across these stages, the level design workflow functions as an *adaptive translation pipeline* that reinterprets algorithmically generated music into interactive, playable, and clinically meaningful material. It operationalizes the design principles of accessibility, flexibility, and extensibility by coupling intuitive authoring methods with direct evaluation in real gameplay contexts.

The specific interface design, editing workflow, and configuration procedures for implementing this level design process are detailed in Chapter 6.

5.5. Acknowledgement

This chapter was entirely drafted and written by the author. ChatGPT was used only for phrasing and minor language refinement. All ideas, analysis, and arguments are the author's own. Prompts were like: help me to check the grammar and refine the content.

6

Implementation

6.1. System Overview

Figure 6.1 presents an overview of the complete pipeline connecting symbolic music generation, level authoring, and interactive execution in **PIZZICATO**. The system was designed as a modular, end-to-end workflow that transforms algorithmically generated musical material into gesture-based rehabilitation tasks, while maintaining consistent data exchange across all stages.

The workflow unfolds across four interconnected modules:

1. **Music Generation.** Users access a web-based interface built around the *Whole-Song Hierarchical Generation* model [**whole-song-hierarchical-diffusion**]. Through visual configuration of musical parameters—such as key, tempo, and section layout—the system generates structured, multi-track MIDI compositions. The generated song is accompanied by a metadata file describing sectional form and timing boundaries.
2. **Level Authoring: Music Notes Editing.** The generated symbolic music is opened in the first page of the Level Authoring module, implemented in HTML and JavaScript. Here, users can inspect, modify, and simplify the musical material through note-level operations. The interface supports both *normal* and *reduced* versions of the music and employs a *trigger–support note* mechanism to group notes, reducing interaction density while preserving musical continuity.
3. **Level Authoring: Target Configuration.** The second page of the Level Authoring module bridges notes editing and real gameplay design. It translates trigger notes into interactive targets that can be spatially arranged and linked to specific pinch gestures. A synchronized playback window provides a real-time preview of how notes appear on the game canvas within the game timeline. Users can manually position targets and assign fingers before exporting the finalized configuration as a JSON file.
4. **Integration with PIZZICATO Runtime.** The exported configuration files can then be loaded into the PIZZICATO game environment, where they define the timing, layout, and gesture logic of each interactive level. Users can adjust global parameters such as tempo (BPM) and progression thresholds and review their performance through automatically generated CSV logs. Additional improvements were implemented to enhance player feedback—specifically, maintaining finger-color consistency by replacing hue shifts with brightness changes when timing cues are reached or missed.

Each stage outputs standardized data formats—MIDI, JSON, and CSV—ensuring seamless interoperability between modules. The entire pipeline is web-based: the generator operates on a remote server through a public port, while the authoring and runtime modules are hosted on GitHub Pages, allowing full accessibility without local installation. Together, these components form a reproducible and extensible ecosystem that connects algorithmic composition with embodied, gesture-driven interaction of sonification game for motor rehabilitation research.

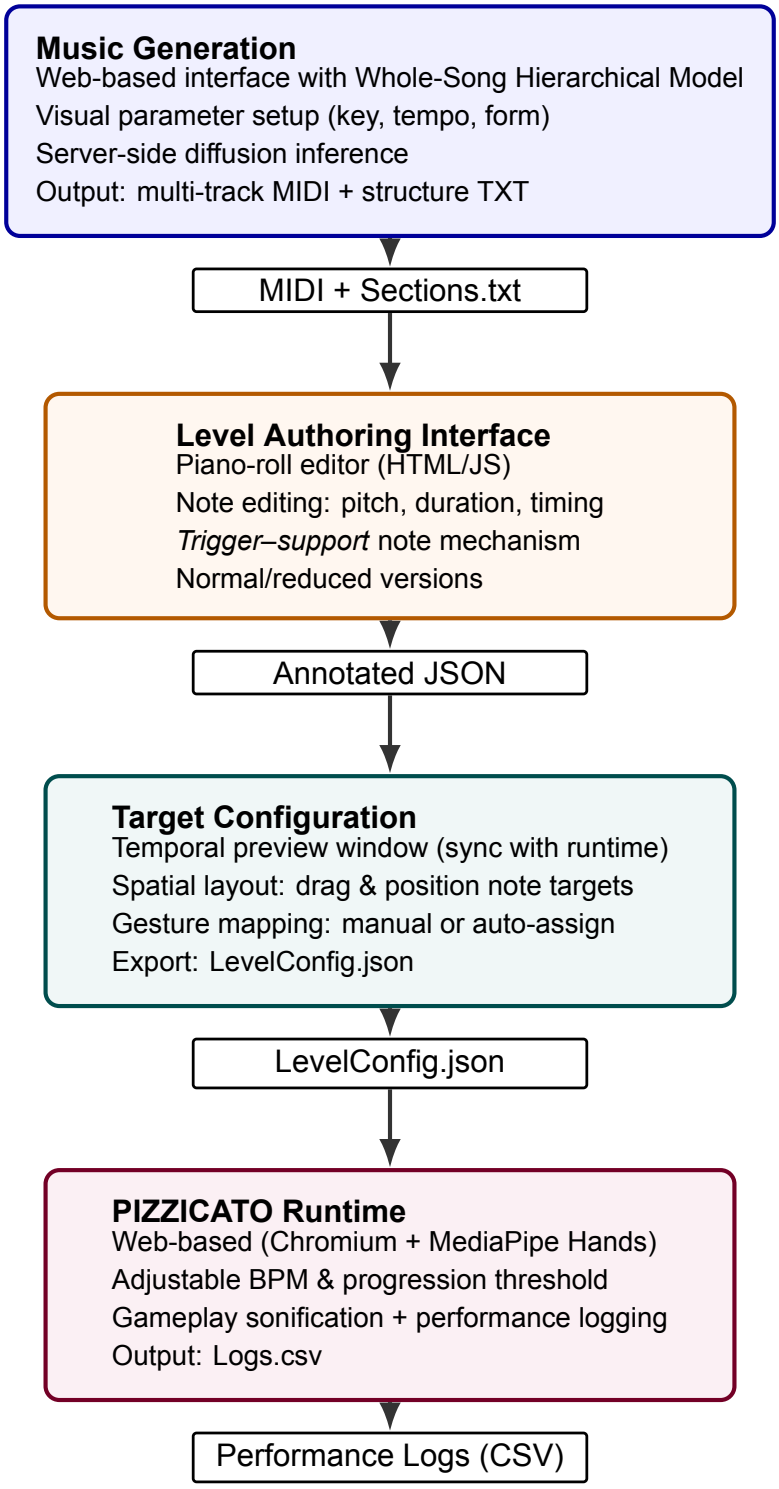

Figure 6.1: System overview of the implemented pipeline. The workflow proceeds from music generation to authoring, target configuration, and runtime integration. Each module outputs standardized artifacts (MIDI, JSON, CSV) ensuring interoperability. The generator operates on a remote server, while authoring and runtime modules are hosted on GitHub Pages for browser-based accessibility.

6.2. Music Generation

Model Adaptation and Simplification

The symbolic music generation process was implemented through a web-based graphical interface that integrates the **Whole-Song Hierarchical Generation** framework [**whole-song-hierarchical-diffusion**]. This model employs cascaded diffusion networks trained on symbolic pop music, capable of generating compositions with coherent global structure and phrase-level organization.

To suit the requirements of sonification-based gameplay, the generation pipeline was modified to operate only up to the *lead sheet* level of the hierarchical model. This ensures that the output captures the essential melodic and harmonic content—adequate for sonification of gestures—while omitting higher-level of accompaniment which contains too many details unnecessary for authoring the game levels. The generated lead sheets, containing both melody and chord tracks, can be directly downloaded and imported for further level design and sonification.

Model configuration

To lower the learning barrier for non-technical and non-musician users, the system introduces an intuitive web-based interface [1] that enables direct interaction with the music generation algorithm as shown in Figure 6.2 . Instead of exposing model parameters in abstract numerical or technical form, the interface translates them into visually meaningful elements—such as labeled blocks, tempo input box, and key selectors—that communicate the underlying musical logic directly. These visual explanatory components were intentionally designed to make the generative process transparent and self-explanatory. Therefore, through this web interface, users such as therapists and researchers can intuitively explore different musical configurations then generate structured symbolic pieces for gesture-based sonification game level authoring without the need for programming or music production expertise.

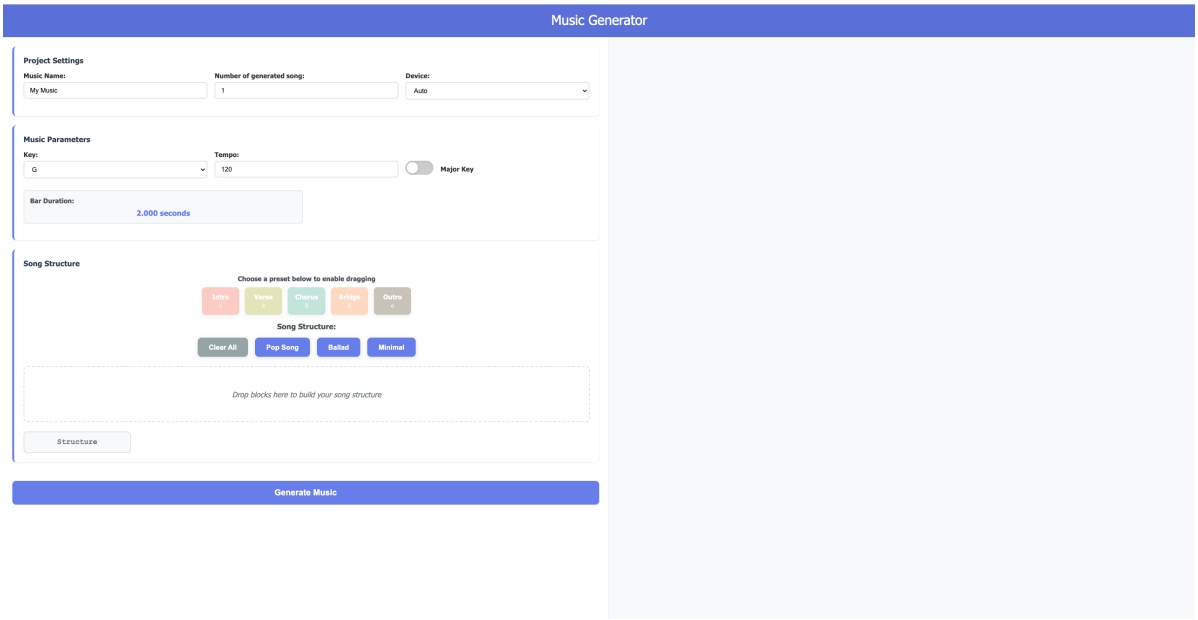

Figure 6.2: User interface of the **Music Generator** module. Users can configure musical parameters, select or drag section blocks to design song structure, and generate complete symbolic compositions through the hierarchical diffusion model.

The interface enables users to define a set of high-level musical attributes through three grouped panels: *Project Settings*, *Music Parameters*, and *Song Structure*, as illustrated in Figures 6.3–6.5. Each panel was designed to translate abstract model parameters into perceptually meaningful controls, ensuring that users without technical or musical expertise can understand the meaning through clear visual cues.

In the **Project Settings** panel (Figure 6.3), users specify the name of the composition, the number of pieces to be generated, and the preferred computing device (CPU or GPU). This panel encapsulates

[1] `http://131.180.119.125:5001/`

the essential project-level configuration, deliberately kept minimal to prevent cognitive overload.

The **Music Parameters** panel (Figure 6.4) provides controls for musical key, tempo, and mode selection. The key can be set to any pitch class (A–G), and the toggle for *Major Key* switches between major and minor scales. A calculated bar duration (in seconds) is displayed dynamically according to the selected tempo, giving users temporal awareness of the speed of the generated song. By presenting abstract musical properties through simple numerical values, this panel enables users without formal musical training to configure fundamental generation parameters intuitively and to clearly interpret how their adjustments influence the temporal and tonal character of the resulting composition.

The most distinctive element is the **Song Structure** panel (Figure 6.5), which provides an intuitive block-based editor for designing the overall form of the song. Predefined templates (*Pop Song*, *Ballad*, or *Minimal*) automatically populate common structural patterns that serve as starting points. Users can then drag and arrange length of color-coded section blocks—*Intro*, *Verse*, *Chorus*, *Bridge*, and *Outro*—to compose their desired structure based on the selected template. This design allows users to configure the structure of the generated song interactively, eliminating the need for manual notation or technical terminology. By presenting the compositional process as a spatial arrangement of blocks, the interface supports direct manipulation and promotes immediate comprehension of overall musical form.

Once satisfied with the configuration, the user clicks the *Generate Music* button, which triggers the hierarchical diffusion process. The backend then proceeds through the three diffusion stages—structural planning, arrangement generation, and note-level realization—ultimately returning a complete symbolic composition.

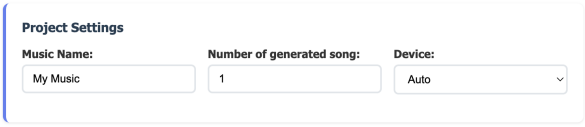

Figure 6.3: The **Project Settings** panel, where users define the project name, generation count, and computing device.

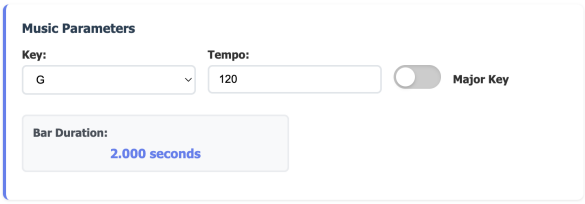

Figure 6.4: The **Music Parameters** panel, showing controls for musical key, tempo, and mode selection, with dynamically calculated bar duration.

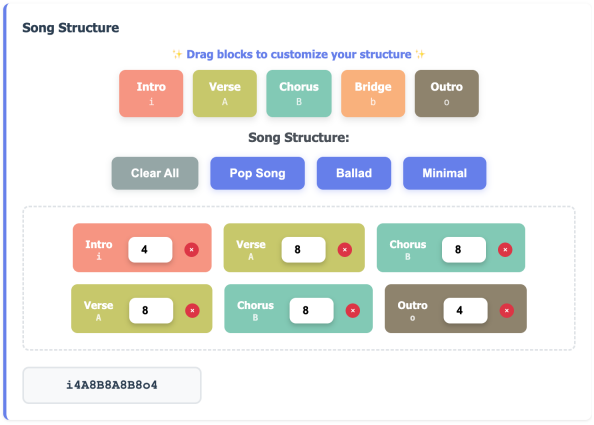

Figure 6.5: The **Song Structure** panel, providing a block-based editor where users can modify section blocks after applying predefined templates to design the song form.

Post-processing and export

During generation, the interface visualizes the progress of the hierarchical diffusion process across its major stages—*Initialization*, *Form Generation*, *Counterpoint*, *Lead Sheet*, and *Finalization*—as shown in Figure 6.6. Each stage provides real-time visualized feedback on the current step, enabling users to understand the procedural flow and monitor the progress of generation.

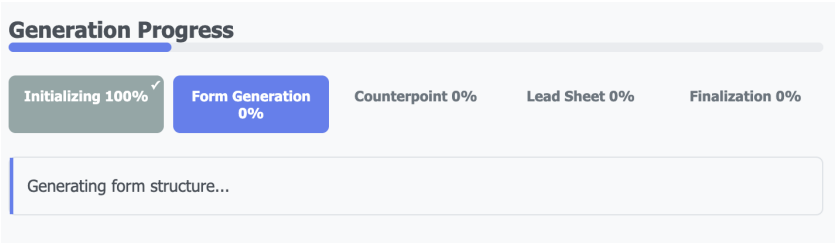

Figure 6.6: Visualization of the hierarchical diffusion process of successive generation stages.

After generation, the system packages the results into a downloadable ZIP archive containing three core artifacts, as illustrated in Figure 6.7. Users can choose to download all outputs as a single compressed archive contains:

- **MIDI file:** a multi-track symbolic representation of the generated composition, including melody, chords, and reduced version of them;
- **Form file (TXT):** a textual summary of the song structure, indicating the sequence of sections such as Intro, Verse, and Chorus;
- **Description file (TXT):** a lightweight metadata file containing configuration details and model parameters used during generation.

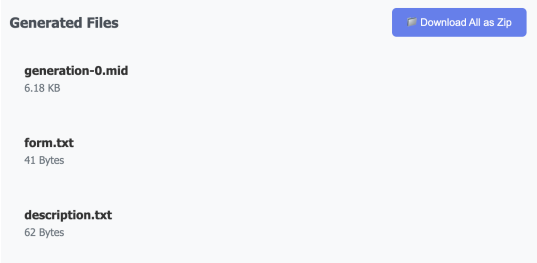

Figure 6.7: Export interface showing the generated files can be downloaded as a ZIP archive.

6.3. Level Authoring

To enable users to configure the four core characteristics of each interactive target—(1) its temporal placement (when the target appears), (2) its spatial placement (where it is positioned on the screen), (3) its gesture mapping (which hand or finger movement triggers it), and (4) its auditory mapping (which sound is produced upon a correct action)—the level authoring interface was implemented as a lightweight web application built with *HTML* and *JavaScript*.

Users can switch between the **Notes Edit** and **Note Placement** tabs to either modify and annotate the musical content directly or configure the corresponding interactive targets that will later be played within the gesture-based sonification game within a unified environment. To ensure accessibility for non-technical and non-musician users, the interface [2] translates these abstract mappings into visually and aurally intuitive elements. For example, temporal relationships are illustrated through synchronized timelines, spatial arrangements are displayed on a draggable canvas, and gesture assignments are color-coded according to finger pairings. Moreover, chord notes that share the same onset and offset times are grouped as a single note to reduce visual and operational complexity.

Note Edit

Within this tab, users can open a generated song directly from the exported folder and switch between different musical layers (e.g., *melody* or *chord*) for detailed editing. A toggle allows users to select between the *normal* and *reduced* versions of the music, offering flexibility in adjusting the level of musical simplicity. Users can also open previously annotated songs (saved from earlier editing sessions), allowing them to compose multiple game levels based on the same musical material. All modifications are reflected in real time on the piano-roll canvas, enabling users to preview and validate edits immediately.

Figure 6.8 illustrates the main editing interface, which consists of three panels: (1) a **General Information Panel**, displaying the song name, tempo, and available tracks; (2) a **Display Panel**, containing a progression bar and a piano-roll canvas that visualizes all musical notes, with section-based color coding to provide intuitive structural understanding; and (3) a **Note Inspector**, which presents detailed note attributes such as pitch, timing, and duration, as well as the relationship between trigger and support notes. Together, these components provide users with a clear overview of both the structural and temporal aspects of the music while supporting efficient, iterative editing.

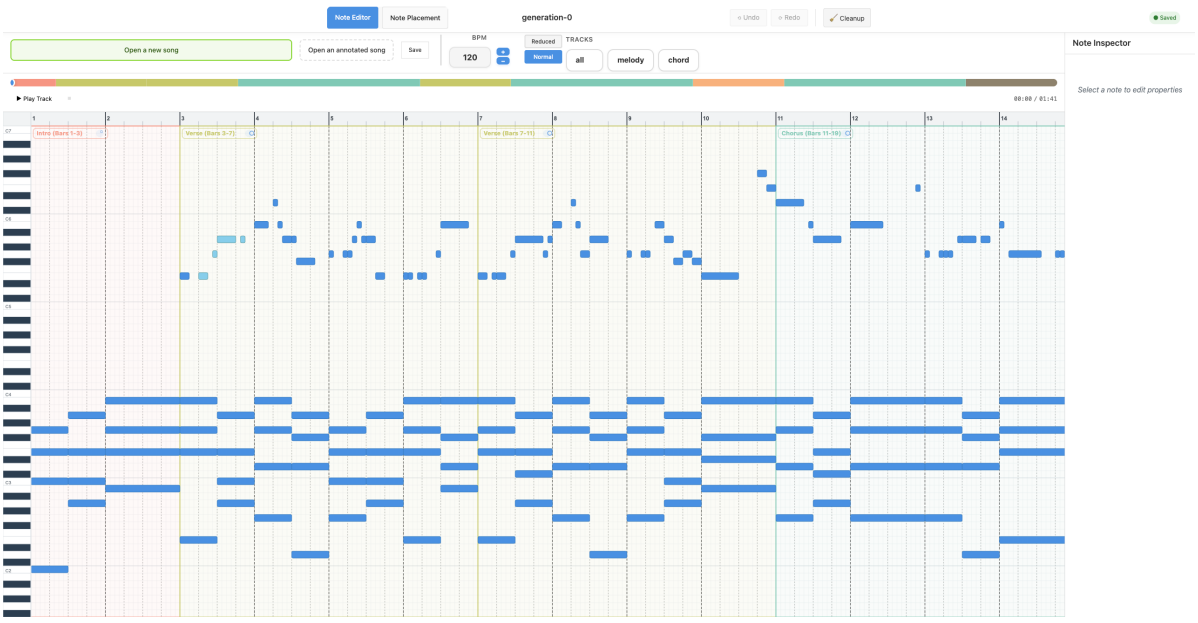

Figure 6.8: User interface of the **Level Authoring Interface**. Users can observe and modify the generated song's details through this page.

[2] `https://qinxinren1.github.io/Level_Designer/`

Music note edit

This is the key function under the note edit page. It allows users to freely modify fundamental note parameters such as *pitch*, *duration*, and *start time*.

The inspector displays timing in intuitive musical units (bars and beats) and shows pitch instead of terminology but simple numbers, allowing non-musician users to understand and control musical parameters without specialized theory knowledge. These design choices—simplified grouping, intuitive visualization, or numeric representation make note-level editing perceptually meaningful, aligning with the system's broader goal of making music editing accessible and transparent to non-technical and non-musician users.

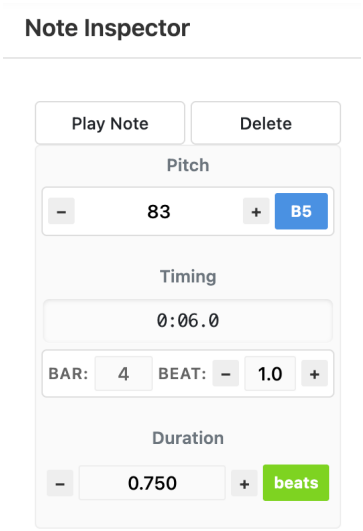

Figure 6.9: Note editing interface. It exposes pitch, timing (bar/beat), and duration controls.

Trigger–support note mechanism

Another key feature of the note editing page is the **trigger–support note** classification (Figure 6.10), which enables users to configure the auditory mapping in greater detail. When a composition is imported, users can adjust the target density by assigning properties to individual notes: *Trigger notes* represent interaction targets that will later be mapped to user gestures, while *support notes* are accompanying tones that will be played sequentially when a trigger note is sonified without requiring additional interaction. If a trigger is missed during gameplay, all associated support notes are skipped as well, maintaining rhythmic and structural consistency.

This mechanism simplifies interaction density by grouping discrete musical notes into coherent phrases, ensuring that each level remains both musically richness and integrity, and motor capabilities feasible for therapeutic use.

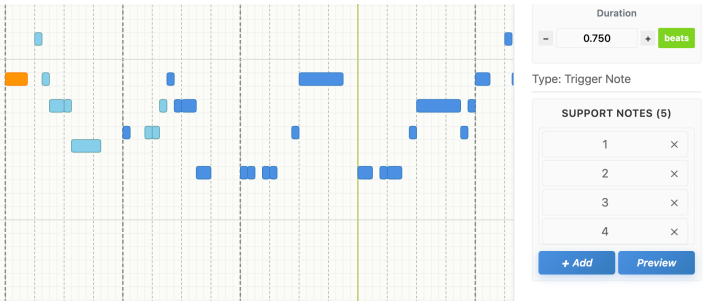

Figure 6.10: Trigger–support mechanism. A selected trigger note shows its associated support notes in the panel (right), enabling phrase-level playback from a single interaction target to reduce motor load while preserving musical continuity.

Target Configuration

After editing the note properties, users proceed to the Target Configuration page, where they define how the musical material is translated into interactive elements within the gesture-based sonification game. This stage primarily addresses the visual and interactive representation of targets—specifically, determining where and when each target should appear, and which gesture should be assigned to trigger its corresponding sound. All configuration processes are designed to be highly intuitive, allowing users to interact directly with visual elements rather than entering numerical parameters manually. A built-in preview mode enables users to immediately observe how their configurations will appear and behave in the actual gameplay environment.

As shown in Figure 6.11, the interface consists of three main components: (1) a time window allows users to freely drag and navigate through different segments of the song to display the corresponding targets. (2) all trigger notes within the selected time window appear on the canvas, where users can freely drag and drop targets or assign specific fingers to them. (3) Users can manually assign or modify finger mappings by clicking the corresponding buttons.

All these features are implemented in a web-based environment, eliminating the need for manual data entry or complex operations. This design not only facilitates rapid iteration but also lowers the barrier to level composition, allowing non-technical and non-musician users to intuitively perceive and refine the relationship between sound and movement.

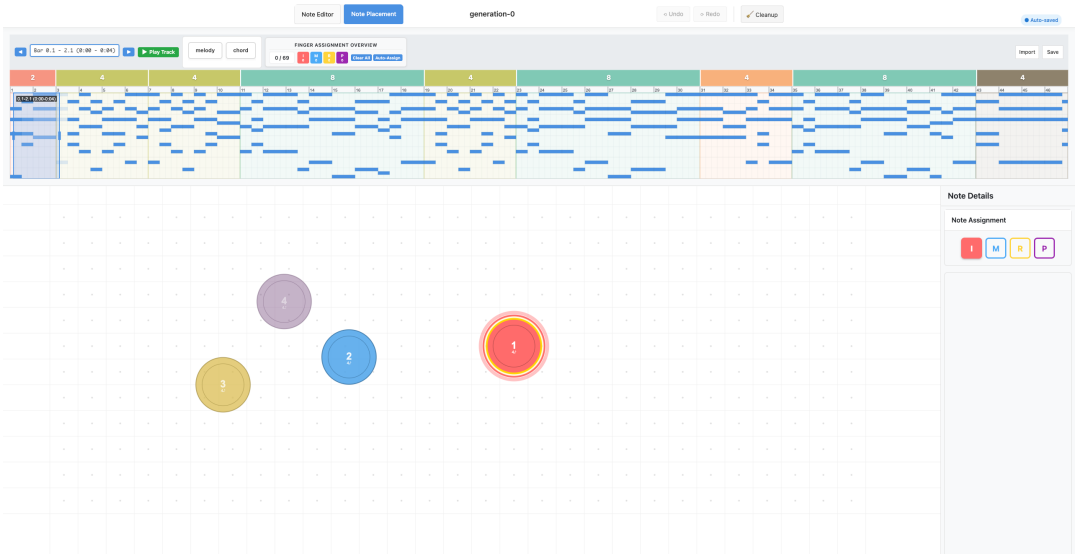

Figure 6.11: Interface of the **Target Configuration** module. The upper piano-roll view shows the generated notes and section structure, while the lower canvas visualizes their corresponding interactive targets. Users can position targets spatially and assign gestures to them directly.

Preview window

To help users understand how game targets unfold over time, the interface includes a **preview time window** synchronized with the real gameplay progress (Figure 6.12). This window defines the visible time range within which note targets are displayed, and its duration matches the preview horizon in the *PIZZICATO* runtime, ensuring consistency between the design and gameplay environments. The left boundary represents the current playback position, while the right boundary marks the upcoming segment of the song. As the timeline scrolls, users can reposition the window to inspect any section of the composition. Only *trigger notes* that fall within this window are rendered in the spatial canvas below, providing a clear temporal focus for editing. These design choices allow users to anticipate the progression of gameplay and understand how the configured targets will appear and behave in practice.

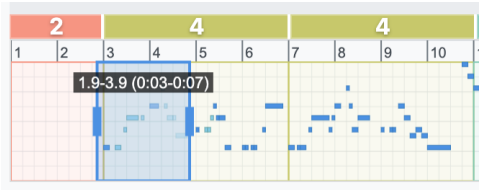

Figure 6.12: Preview time window. The blue frame defines the active time range that will be displayed on the spatial canvas. Its duration matches the preview horizon in the runtime, providing temporal focus and consistency between editing and gameplay.

Spatial layout assignment

The lower canvas visualizes the in-game spatial layout of the note targets. Each active trigger note within the preview window is displayed as a circular node that users can freely drag to any position. By iteratively adjusting positions, users can balance visual clarity, rhythmic flow, and playability, ensuring that each level remains both expressive and physically feasible. This design supports perceptual intuition: rather than configuring coordinates numerically, users directly manipulate spatial relationships through drag-and-drop interaction, which mirrors the embodied nature of gameplay itself.

Gesture assignment

Each spatial target should be linked to a pinch gesture corresponding to a specific finger pair (e.g., thumb–index, thumb–middle). Users can manually assign gestures to individual targets or employ the *auto-assign* function to distribute finger mappings systematically. This integration gives the user flexibility to configure different gestures and patterns dependent on their patient's condition, and then they can preview the targets by moving the time window or start preview mode to ensures that spatial positioning and physical motion remain coherent before the level is finalized and exported for gameplay.

Section overview

As shown in Figure 6.13, when users click on a specific section label, the interface highlights all corresponding targets within that section. This feature provides an immediate visual overview of note density and gesture distribution across different musical segments, helping designers balance complexity and ensure smooth gameplay progression.

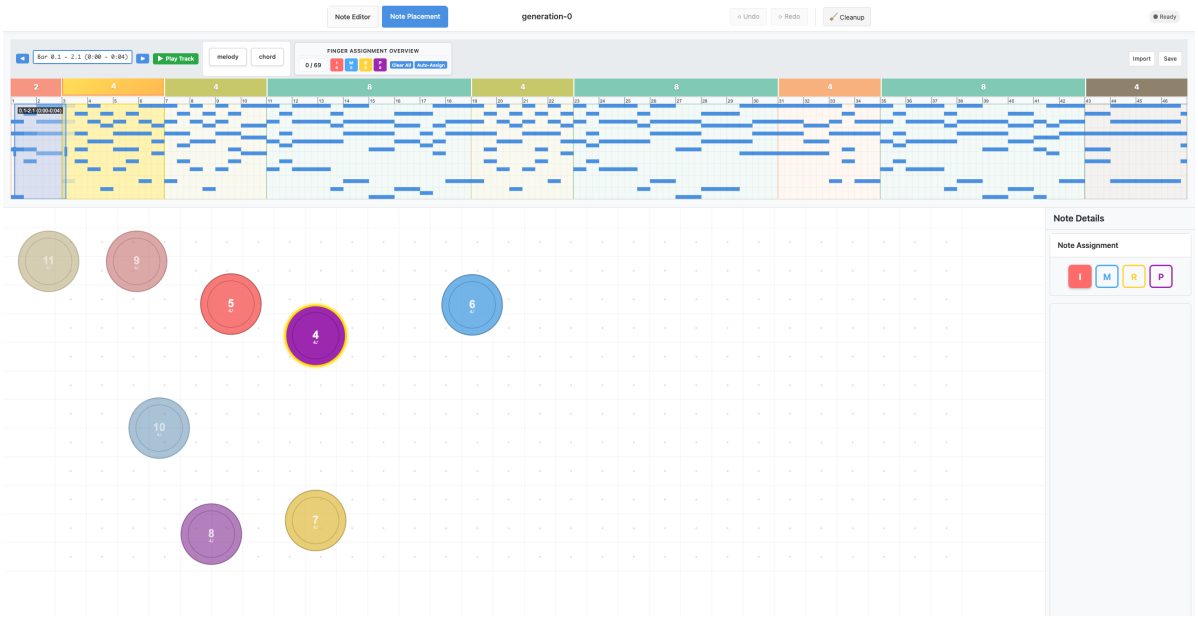

Figure 6.13: Visualization of section-based target selection. When a section (e.g., *Verse* or *Chorus*) is selected, all related note targets are simultaneously highlighted on the spatial canvas. This allows users to inspect and adjust local gesture layouts within the broader temporal and structural context of the piece.

Export configuration

Once finalized, the level is exported as a JSON configuration file containing all necessary parameters: note timings, spatial coordinates, gesture assignments, and section labels. Levels adhere to PIZZICATO's (an instance of gesture-based sonification game) layered architecture, typically starting with rhythm or chord layers and progressively introducing melody layers.

6.4. Integration with PIZZICATO

After authoring the level—defining the timing, spatial layout, and gesture logic for each interactive element—users can test their levels directly within the **PIZZICATO**[3] environment. This integration process allows them to put their designs into practical use and evaluate how the authored content behaves during real gameplay. Several modifications were made to *PIZZICATO* itself to support the uploading and execution of custom levels, ensuring seamless adaptation between the authoring interface and the runtime environment.

Level Upload

Users can then upload their designed level through this interface in Pizzicato directly instead of manually access the Pizzicato's database. All uploaded levels are visible within the interface and can be replayed, edited, or tested iteratively. Figure 6.14 shows the *Level Upload* interface, where users can manage their library of generated levels by uploading new files, playing, editing, or deleting existing ones. The layout was designed to emphasize *transparency and recoverability*, allowing users to confidently experiment and reconfigure levels without risk of losing progress.

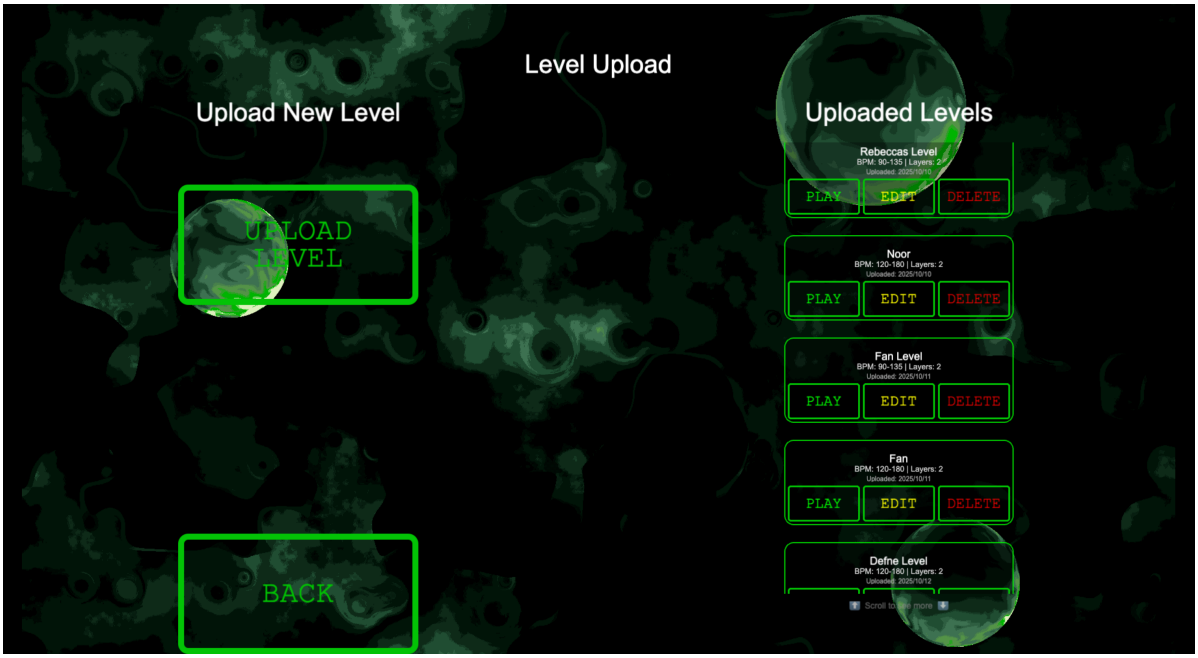

Figure 6.14: Level Upload interface. Users can upload new levels, view previously imported configurations, and perform actions such as play, edit, or delete. Color-coded buttons and clear layout organization enhance visibility and prevent accidental errors.

During **PIZZICATO** gameplay, the runtime synchronizes the symbolic timing of notes with hand-tracking input from a webcam, using MediaPipe's real-time detection framework. When a player performs a correct pinch gesture within the designated time window, the corresponding note is triggered, contributing to the ongoing musical texture. Missed or mistimed notes remain silent, preserving rhythmic continuity while offering implicit feedback on precision. After each session, the system automatically generates a **performance log (CSV)** containing detailed information about onset deviation, hit accuracy, gesture

[3] https://qinxinren1.github.io/Pizzicato_Level_Editor/

distribution, and per-layer progression. These data can later be analyzed to evaluate motor performance and learning outcomes.

Level Settings and Difficulty Calibration

To ensure that the same authored level can accommodate players with different abilities, the runtime offers a dedicated settings interface (Figure 6.15). Here, users can select predefined difficulty levels (*Easy*, *Medium*, *Hard*) that automatically adjust the playback tempo (e.g., 90–150 BPM) and fine-tune the *Progression Count*, which determines how many correct hits are required to unlock subsequent musical layers. This design supports adaptive rehabilitation scenarios where task complexity can be scaled to match individual performance or progression speed. By mapping difficulty to intuitive categories and continuously displaying the corresponding BPM value, the interface maintains perceptual clarity while providing fine-grained control over pacing and challenge level.

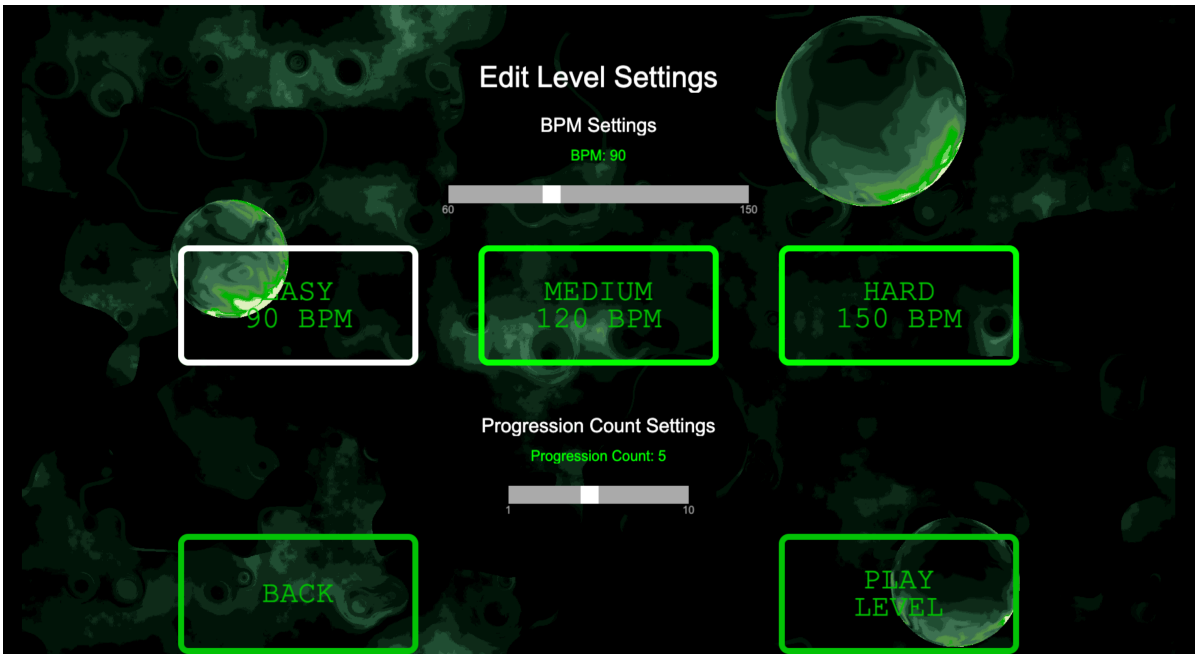

Figure 6.15: Level Settings interface. Users can adjust BPM (tempo) and progression thresholds to modify level difficulty. The clear visual contrast and structured layout facilitate quick calibration and adaptive difficulty adjustment for therapeutic contexts.

System improvements

In the course of this research, several interface-level improvements were introduced to enhance usability and accessibility of the PIZZICATO runtime. In the original implementation, target nodes turned green when the optimal timing moment was reached. However, this temporary color change obscured the original color coding that indicated the assigned finger, causing confusion among players—particularly those performing motor exercises requiring clear finger differentiation. To address this issue, the color feedback system was redesigned: rather than changing hue, targets now adopt a *brighter variant* of their original color when the ideal pinch time is reached. This preserves both timing and finger identity cues simultaneously, reducing cognitive load and improving the intuitiveness of the feedback.

6.5. Technical Deployment

The system components were deployed across lightweight, web-accessible environments to facilitate remote testing, collaboration, and reproducibility. Each module—the generator, editor, and runtime—can be accessed independently through public URLs, ensuring platform flexibility and minimal setup requirements for users and researchers.

Deployment and accessibility

The **Music Generator** module is hosted on a remote server with a publicly accessible port, allowing users to perform symbolic music generation directly through a web interface. The server environment runs a pre-trained diffusion model backend and communicates with the frontend via RESTful APIs. This separation ensures that computationally intensive generation tasks are executed remotely, while users interact only with the lightweight web interface.

The **Level Authoring Interface** and **Target Configuration** modules are client-side applications published on GitHub Pages. They can be accessed directly through a browser without any installation or local setup. All data (e.g., MIDI and JSON files) is stored locally within the browser session and can be exported or re-imported manually by the user. This approach enhances portability and enables therapists or researchers to design, modify, and share levels from any device with modern browser support.

Similarly, the **PIZZICATO Runtime** can be executed through its own GitHub Page, integrating MediaPipe's on-device hand-tracking system. Since gesture recognition and audio playback occur entirely in the browser, the runtime requires no additional software dependencies or hardware beyond a webcam. This low-threshold deployment model aligns with the project's goal of accessibility and reproducibility in therapeutic and research contexts.

Interoperability and workflow

The full workflow—from symbolic generation to interactive testing—is designed for interoperability. Each stage outputs standardized artifacts:

- Music Generator $\rightarrow$ multi-track MIDI + section annotations (JSON)
- Level Authoring $\rightarrow$ edited and gesture-annotated configuration (JSON)
- Runtime $\rightarrow$ performance logs (CSV)

Because each component uses standard web technologies, they can be hosted independently or combined within a shared online workspace. This modular structure enables future extensions such as real-time data streaming, user studies conducted fully online, or integration with external rehabilitation databases.

6.6. Summary

This chapter detailed the practical implementation of the end-to-end system connecting symbolic music generation, level authoring, and interactive rehabilitation gameplay in *PIZZICATO*. The design follows a modular and web-based architecture, ensuring accessibility, interoperability, and reproducibility across all stages of the workflow.

The **Music Generation** module employs a hierarchical diffusion model to produce structured, multi-track symbolic compositions from high-level user input. These outputs are then refined within the **Level Authoring Interface**, where users can edit musical content, simplify notes density by applying the trigger–support note mechanism to adapt musical complexity to therapeutic requirements. The **Target Configuration** module extends this process into spatial and gestural domains, allowing users to map musical notes onto visual targets and assign specific finger gestures while previewing real-time gameplay behavior. Finally, the **PIZZICATO Runtime** integrates these configurations into the interactive game environment, translating symbolic structure into embodied, gesture-based sonification and recording detailed player performance data.

Together, these components form a coherent pipeline that bridges algorithmic composition with motor-skill sonification. By automating the transition from symbolic music to interactive rehabilitation levels, the system empowers non-musician researchers and therapists to design musically meaningful and physiologically feasible training tasks. The following chapter evaluates this implementation in practice, focusing on usability, accessibility, and the quality of interaction feedback experienced by test participants.

6.7. Acknowledgement

This chapter was entirely drafted and written by the author. ChatGPT was used only for phrasing and minor language refinement. All ideas, analysis, and arguments are the author's own. Prompts were like: help me to check the grammar and improve the phrasing.

7

Evaluation

7.1. Evaluation Objectives

The purpose of this evaluation is to determine to what extent the proposed authoring pipeline meets the needs of therapists and researchers who lack technical or musical expertise. Specifically, the study examines the usability, flexibility, and learnability of both the music generation interface and the level editor, assessing whether users can independently produce musical material and translate it into playable gesture-based levels as defined in the design principles earlier. In addition, the evaluation investigates how effectively the end-to-end workflow—from music generation to in-game deployment—supports non-technical and non-musician users in carrying out their intended tasks without external assistance. Finally, expert feedback from rehabilitation professionals and researchers is collected to understand how well the system integrates with real therapeutic or experimental workflows and to identify opportunities for refinement and future development.

7.2. Methodology

7.2.1. Participants

For the assessment, we recruited a total of nine testers. All participants were psychologists or therapists with professional experience in studying the impact of music on motor behavior, particularly in the context of gesture-based sonification games used for patients with motor skill disabilities. Their domain expertise makes them well-suited to evaluate the system's usability and relevance for therapeutic and research applications.

PIZZICATO is actively used in their clinical research, and participants expressed a clear need for a usable and flexible way to design their own musical levels tailored to the needs of their patients. However, they typically lack programming expertise, and most do not have sufficient musical background to compose or prepare suitable musical material on their own. These constraints highlight the importance of an accessible authoring pipeline that allows non-technical and non-musician clinicians to generate music and configure gesture-based sonification levels efficiently.

7.2.2. Study Design

Given the limited number of participants, this evaluation adopts a qualitative study design rather than a quantitative one. The procedure consisted of three main stages. First, participants received a brief explanatory introduction to the system, including an overview of the music generation interface, the level editor, the target configuration, and the integration with the *Pizzicato* workflow. Second, they were invited to freely explore and interact with the interfaces while thinking aloud, during which they were encouraged to ask questions or report difficulties whenever they encountered confusion or obstacles. This phase allowed us to observe their natural interaction patterns and identify usability issues. Third, each session concluded with a semi-structured interview aimed at gathering in-depth feedback on their experience, focusing on clarity, usability, flexibility, and perceived applicability of the system in therapeutic or research contexts.

7.2.3. Tasks

Participants completed the following tasks during the evaluation session:

1. Configure the parameters needed to generate a piece of music using the hierarchical music generator.
2. Open the generated song in the level editor and edit the musical notes using the trigger–support note mechanism.
3. Configure gesture assignments and spatial layout for the annotated musical material in the target configuration interface.
4. Export the designed level, load it into PIZZICATO, and play through the level to evaluate its behavior during gameplay.

7.2.4. Data Collection

The evaluation relied on qualitative data sources to capture users' perceptions, challenges, and interaction patterns. Two primary methods were used:

- **Qualitative observations**: During the think-aloud interaction phase, we recorded participants' behaviors, comments, questions, and difficulties to identify usability issues and interaction bottlenecks.
- **Post-session interviews**: Semi-structured interviews were conducted at the end of each session to gather in-depth feedback on the usability, clarity, and practical relevance of the music generation tool, level editor, and target configuration workflow.

7.2.5. Evaluation Setup

The evaluations were conducted in both offline and online formats. For the offline sessions, participants interacted with the system in person, and their performance and feedback were captured through audio recordings while we took observational notes. For the online sessions, participants accessed the music generator, level editor, and PIZZICATO runtime through web links. During these remote evaluations, visual recordings of their screen interactions were collected to document their performance, identify usability challenges, and support subsequent qualitative analysis.

7.3. Results

7.3.1. Usability of Music Generator (O1)

Across participants, clear differences emerged between users with and without prior musical knowledge. Those with musical experience were able to configure the generator parameters quickly and intuitively. However, participants without a musical background also reported that the interface felt accessible, largely due to the explanatory hints provided for key parameters. In particular, parameters such as tempo (BPM) and key were viewed as necessary and understandable within the context of their therapeutic work.

Participants expressed satisfaction with the generated musical output. The two provided layers—melody and chords—were consistently described as sufficient for creating levels appropriate for their patients. While some participants noted that the automatically generated pieces were slightly long for therapeutic sessions, they also acknowledged that the subsequent editing interface allows them to trim sections easily, making the length acceptable in practice.

The structural control offered by the section-based configuration (intro, verse, chorus, outro) was also positively received. Testers appreciated the availability of predefined structural templates, which gave them a meaningful starting point. At the same time, the ability to freely modify these templates allowed enough flexibility to adapt the musical form to their specific therapeutic needs.

7.3.2. Usability of Level Editor (O2)

Notes Editing

In this stage of the evaluation, testers were asked to modify the generated song according to their therapeutic or experimental needs. The editing operations included adjusting individual note properties

such as pitch, duration, and onset time, as well as trimming unwanted sections to shorten the musical piece.

Participants reported that these editing functions were simple yet sufficient for their purposes. They appreciated being able to make small adjustments without requiring advanced musical knowledge, and emphasized that the interface avoided overwhelming them with complex compositional controls. The interaction style based on clicking and dragging notes was described as intuitive and easy to use, allowing them to perform the necessary modifications with minimal learning effort.

Trigger - Support Mechanism

Therapists and psychologists reported that in the original version of PIZZICATO, many levels were too difficult for patients with limited motor skills to perform comfortably. As a result, they expressed strong appreciation for the trigger–support mechanism, which groups multiple musical notes into a single interactive target, thereby significantly reducing the number of required gestures while still preserving the musical information and rhythmic structure of the piece. Participants described the mechanism as intuitive and effective, noting that it enabled them to tailor the interaction density to the needs of different patient groups.

Although the concept was initially unfamiliar, testers quickly understood how it worked and were able to apply it confidently. The process of assigning trigger notes and support notes was perceived as smooth and efficient, and participants emphasized that it was well-suited for users without technical or musical expertise.

7.3.3. Target Assignment

Time Window for Preview

In this component of the interface, testers were asked to manipulate the preview time window by dragging it along the timeline to observe how targets would appear on the game canvas. Participants found this feature particularly valuable, as it provided an immediate visual preview of the sequence and timing of targets without requiring them to export the level to the actual game. This substantially lowered the barrier for iteration and made level design more efficient and accessible.

Although the concept of a movable preview window was initially unfamiliar to most testers, they quickly understood its purpose after a brief explanation. Once familiar, they were able to operate it easily and reported that it greatly supported their understanding of how the designed level would behave during gameplay.

Spatial Layout and Finger Assignment

Testers configured the spatial layout of targets by dragging and dropping them on the game canvas. They reported that this interaction felt intuitive and easy to use, and that the snap-grid mechanism was particularly helpful when arranging targets into structured patterns. The ability to highlight and display all targets within a specific musical section by clicking the corresponding section block was also regarded as highly valuable. This feature enabled participants to design spatial patterns tailored to the abilities or therapeutic goals of individual patients, making the layout design process more deliberate and clinically meaningful.

Regarding finger assignment, participants emphasized that the ability to enable or disable specific fingers was essential for clinical applications, where patients may have different mobility limitations or rehabilitation goals. Both interaction modes—using the keyboard shortcuts or clicking the finger buttons with the mouse—were described as straightforward and convenient. Testers noted that this flexibility allowed them to quickly configure gesture mappings appropriate for various patient conditions.

7.3.4. Integration With PIZZICATO Runtime (O3)

After completing the level design, testers exported their configurations and loaded them into the PIZZICATO runtime to evaluate the level in an actual gameplay scenario. Participants reported that the process of importing levels and adjusting global parameters—such as the tempo (BPM) and the progression threshold—was straightforward and essential for their clinical use cases. When a designed level appeared too challenging, they could easily slow down the BPM or lower the threshold requirement, allowing rapid fine-tuning without needing to return to the editor.

Testers also expressed satisfaction with the updated visual feedback mechanism in PIZZICATO. In the modified version, target nodes transition to a lighter shade of their original color rather than turning green when the optimal timing moment is reached. Participants noted that this improvement preserves the association between color and finger assignment, helping patients focus on motor skill execution rather than relying on memory of previous color changes. Overall, the runtime integration was perceived as smooth, intuitive, and supportive of iterative therapeutic design.

7.3.5. Challenges and Future Work

During the expert walkthroughs and preliminary usability evaluations, several challenges were identified that point to important directions for future development.

1. **Accessibility of the music generation interface.** Although the system was designed for non-musician users, several testers without musical background reported difficulty understanding the meaning and implications of parameters in the Music Generation interface. This suggests that additional scaffolding—such as contextual explanations or an adaptive “beginner instructuin”—may be needed to lower the cognitive barrier for first-time users.
2. **Long generation time.** Users noted that the symbolic music diffusion model required longer-than-expected inference time, especially when generating full-length pieces. This limitation is primarily infrastructural rather than conceptual; deploying the model on a more capable server or implementing asynchronous generation with progress feedback could significantly improve user experience.
3. **Limited freedom in music selection.** Several testers expressed the desire to import their own musical material rather than relying exclusively on generated content. Supporting user-uploaded symbolic music not constrained the the format of generated music would enhance creative flexibility and make the system more applicable to therapeutic contexts where familiar or patient-preferred music is beneficial. Future work may explore integrating automatic preprocessing or quantization pipelines to ensure that uploaded music remains compatible with the sonification and level-authoring workflow.
4. **Need for tutorial material.** Testers indicated that a dedicated instruction guidebook or tutorial video would substantially improve their understanding of the interface and reduce onboarding time. Including step-by-step examples or embedded help widgets within the interface would further support independent learning and reduce reliance on the researcher during initial use.
5. **Lack of clinical testing and adaptive difficulty requirements.** Although domain experts were able to create complete levels using the system, they have not yet deployed these levels with real patients. As a result, the suitability of difficulty settings, gesture complexity, and temporal density has not been validated in clinical practice. Future work should therefore include pilot studies with patients to evaluate therapeutic effectiveness and inform refinements, such as personalized difficulty scaling or dynamic adaptation mechanisms.

Taken together, these findings highlight the need for both usability improvements and broader integration with therapeutic workflows. Addressing these challenges will help ensure that the authoring system becomes a practical, flexible, and clinically meaningful tool for gesture-based sonification research and rehabilitation.

8

Conclusion

8.1. Summary of Contributions

This thesis presented an end-to-end authoring pipeline that enables non-technical and non-musician users—specifically therapists and psychologists—to design customized levels for gesture-based sonification games such as PIZZICATO. The work addressed challenges related to generating musically coherent material, editing and simplifying symbolic music for motor-rehabilitation use, and configuring gesture-based interactions through an intuitive interface.

First, we introduced a symbolic music generation method based on hierarchical diffusion models, allowing users to create structured, multi-layer musical material without requiring compositional expertise (O1). Second, we developed an accessible level editor and target configuration interface that supports note editing, trigger–support grouping, spatial layout design, and finger assignment, enabling flexible and clinically meaningful level creation (O2). Finally, we evaluated the system with domain experts and demonstrated that the pipeline effectively supports therapeutic workflows while remaining usable for non-musician and non-programmer users (O3).

8.2. Answer to the Research Question

The main research question of this thesis asked:

> *How can tools and methods be designed and integrated to enable non-technical, non-musician users—such as therapists and researchers—to create and customize levels for gesture-based sonification games?*

To answer this question, the study examined two complementary domains introduced in Chapter 1: (1) acquiring suitable musical material, and (2) transforming that material into playable, therapeutically meaningful levels. Below, each sub-research question (RQ1–RQ5) is addressed individually, followed by an integrated conclusion.

RQ1: What musical characteristics make a piece suitable for gesture-based sonification games?

Across Chapters 4 and 6, the study found that gesture-based sonification imposes strict musical requirements because every gesture must correspond to a discrete, perceptually clear note onset. The following characteristics are essential:

- **Clear rhythmic pulse**, supporting anticipatory timing and motor coordination.
- **Separable musical layers**, so that melody, harmony, and rhythm remain perceptually distinct.
- **Salient, temporally precise onsets**, enabling reliable mapping between note events and gesture timing.

Experiments with audio-based processing (e.g., source separation and onset detection) showed that

timing noise and polyphonic ambiguity make such materials unsuitable for rehabilitation-oriented sonification. Symbolic formats (e.g., MIDI) provide the temporal precision required for motor-timed interaction.

Conclusion (RQ1): Suitable musical material must contain clean, discrete, structurally coherent note events with minimal onset ambiguity. This requirement directly motivated the use of symbolic generation.

RQ2: How can non-musician and non-technical users obtain such music without musical expertise?

The study concluded that users must work with tools that *hide compositional complexity* while still allowing control over therapeutically relevant musical features. Audio-to-MIDI transcription proved unsuitable because errors arising from polyphony and expressive timing require expert correction.

Hierarchical symbolic music generation satisfies these needs by offering:

- **High-level controls** (e.g., tempo, length, section layout),
- **Perceptually meaningful interactions** instead of theoretical terminology,
- **Internally managed musical structure** that preserves coherence without user intervention.

Conclusion (RQ2): Non-musical users can acquire suitable material when tools provide intuitive, high-level controls while internally managing low-level musical decisions by algorithms.

RQ3: Which interface features help users translate musical structures into interactive levels intuitively?

Chapter 5 established that level authoring becomes intuitive when users operate through perceptual metaphors rather than numerical or symbolic editing. Three interface principles were essential:

- **Notes Edit** gives users ability to adjust the generated music according to their needs.
- **Direct manipulation interfaces** enable users to drag, place, and assign gestures to targets visually.
- **Spatial–temporal preview windows** allow users to see how targets unfold during gameplay.

These features allow users to reason visually rather than musically or technically, making level composition accessible even without domain expertise.

Conclusion (RQ3): Intuitive level authoring is achieved when musical structures are presented as manipulable visual and temporal elements aligned with gameplay perception.

RQ4: What flexibility and control is needed for therapeutic customization?

Therapeutic contexts vary widely, so users must be able to adjust the difficulty and interaction demands of a level. Two forms of control were found essential:

- **Trigger–support note simplification** reduces motor load by grouping notes while preserving musical expression.
- **Editing and trimming of generated music** enables users to remove or modify undesired notes or sections, effectively shaping the level's complexity.
- **Level speed adjustment** for the uploaded level in Pizzicato allows user to further adjust the difficulty by making the level play faster or slower.

These mechanisms allow users to tailor rhythmic density and interaction frequency without musical expertise.

Conclusion (RQ4): Therapeutic customization requires flexible controls that let clinicians adapt difficulty by note density and speed to individual patient abilities.

RQ5: How can users test and validate their levels to ensure playability and therapeutic appropriateness?

The study found that real-time, in-context testing is essential. Validation requires:

- **Execution within the actual sonification game**,
- **Immediate visual and auditory feedback** in the real game,
- **Rapid iteration** between editing and testing.

PIZZICATO provides this workflow by allowing users to upload and play their authored levels directly. The library of uploaded levels enables users to adjust the difficulty without going back to the authoring page.

Conclusion (RQ5): Effective validation depends on seamless integration with the runtime environment, enabling real-time testing and structured feedback. Pizzicato is examined to be helpful in validation.

8.3. Discussion

The evaluation results indicate a high level of satisfaction among the testers, suggesting that the proposed workflow aligns well with their needs in clinical and research contexts. From a **usability** perspective, participants emphasized that they were able to design complete levels from scratch without requiring prior musical or technical expertise. The music generation module and the intuitive interaction methods—such as clicking, dragging, and immediate visual previews—provided them with sufficient freedom to create custom therapeutic content. These interaction techniques effectively reduced cognitive load and supported rapid iteration, allowing users to explore ideas and refine designs with minimal friction.

Flexibility also emerged as a key factor for therapists and psychologists, particularly given the diverse motor capabilities of their patients. Participants highlighted the importance of being able to adjust task difficulty, and they found the trigger–support mechanism especially valuable in this regard. By preserving musical information while reducing the density of required gestures, this mechanism allowed clinicians to tailor the complexity of levels to specific therapeutic scenarios without compromising musical or rhythmic coherence.

Finally, the **integration** of authored levels within the PIZZICATO runtime provided strong evidence that the overall workflow is practically viable for therapeutic use. Testers were able to import their levels, adjust global parameters such as tempo and threshold, and evaluate gameplay performance seamlessly. Their ability to use the system end-to-end confirms that the workflow is not only functional but well-suited for real-world clinical application.

More broadly, this work demonstrates the potential of combining symbolic music generation with interactive editing tools in rehabilitation-focused sonification systems. By prioritizing usability, flexibility, and clinical relevance, the system bridges the gap between research prototypes and the practical needs of clinicians. The findings suggest that gesture-based sonification can be made substantially more accessible and impactful when supported by dedicated, domain-appropriate authoring environments.

8.4. Limitations and Future Work

This research presents several limitations. First, the expert evaluation involved only nine participants, which restricts the generalizability of the findings and prevents drawing quantitative or statistically supported conclusions. While the qualitative insights were rich and domain-relevant, a larger and more diverse participant group would be necessary to validate the results more broadly.

Broader conceptual limitation of this work is that the role, use, and therapeutic importance of music selection has not been fully theorized, designed, or evaluated within the level-creation pipeline for gesture-based sonification games. The current system builds upon a symbolic generation model trained predominantly on contemporary pop music datasets, which results in generated material that inherits the structural and stylistic conventions of pop. While many experts responded positively to the musical outcomes, several also highlighted that alternative musical forms—particularly classical pieces, shorter excerpts, or culturally familiar melodies—may be more appropriate for stroke rehabilitation or

motor-impaired patients. Older adults, for example, often prefer classical or traditional music, which can reduce cognitive load and increase emotional engagement. Because this research does not yet investigate how different musical genres, forms, or lengths influence therapeutic suitability, future work should systematically explore a broader repertoire of musical inputs beyond pop-derived structures and enable level creation based on diverse musical sources.

A second conceptual limitation lies in the current role of automation in the level authoring process. Although the system provides flexible direct manipulation tools, it does not yet incorporate clinical knowledge—such as empirically supported motor-exercise patterns, recovery progressions, or gesture sequencing strategies—into automated suggestions. In therapeutic contexts, the design of movement tasks is not arbitrary: psychologists and physiotherapists rely on established models of motor learning, repetition structures, and difficulty shaping. The present framework does not integrate these domain insights, meaning authors must manually translate therapeutic intentions into spatial and temporal patterns. Future research should therefore investigate how clinical expertise can be embedded into automated or semi-automated authoring features, guiding users toward movement patterns known to support motor recovery while reducing workload and improving consistency across designed levels.

Additionally, the evaluation presented in this thesis did not involve patient testing, meaning the long-term therapeutic effectiveness of the designed levels remains unassessed. A comprehensive clinical evaluation involving real patients would be essential to understand how the generated and authored levels influence motor skill development, engagement, and rehabilitation outcomes. Future work should therefore include longitudinal studies in clinical environments to measure the practical impact of the system on therapeutic progress.

References

[1] Fernanda M. Alfieri et al. “Gamification in Musculoskeletal Rehabilitation: A Narrative Review”. In: *International Journal of Environmental Research and Public Health* 19.24 (2022), p. 16465. DOI: 10.3390/ijerph192416465. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC9789284/.

[2] Frédéric Bevilacqua et al. “Sensori-Motor Learning with Movement Sonification: Perspectives from Recent Interdisciplinary Studies”. In: *Frontiers in Neuroscience* 10 (2016), p. 385. DOI: 10.3389/fnins.2016.00385. URL: https://www.frontiersin.org/articles/10.3389/fnins.2016.00385/full.

[3] Sara Cardinale, Michael Cook, and Simon Colton. “AI-Driven Sonification of Automatically Designed Games”. In: *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. Vol. 18. 1. 2022, pp. 95–101.

[4] John Dyer, Paul Stapleton, and Matthew Rodger. “Mapping Sonification for Perception and Action in Motor Skill Learning”. In: *Frontiers in Neuroscience* 11 (2017), p. 463. DOI: 10.3389/fnins.2017.00463. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC5566964/.

[5] John Dyer, Paul Stapleton, and Matthew Rodger. “Sonification as Concurrent Augmented Feedback for Motor Skill Learning and the Importance of Mapping Design”. In: *The Open Psychology Journal* 8 (2015), pp. 192–202. DOI: 10.2174/1874350101508010192. URL: https://openpsychologyjournal.com/VOLUME/8/PAGE/192/.

[6] John Dyer, Paul Stapleton, and Matthew Rodger. “Transposing Musical Skill: Sonification of Movement as Guided Action”. In: *Frontiers in Psychology* 7 (2016), p. 1067. DOI: 10.3389/fpsyg.2016.01067.

[7] Alfred O. Effenberg et al. “Movement Sonification: Effects on Motor Learning beyond Rhythmic Adjustments”. In: *Frontiers in Neuroscience* 10 (2016), p. 219. DOI: 10.3389/fnins.2016.00219. URL: https://www.frontiersin.org/articles/10.3389/fnins.2016.00219/full.

[8] Emma Frid and Roberto Bresin. “Interactive Sonification of Spontaneous Movement of Children: Cross-Modal Mapping and Expressive Play”. In: *Frontiers in Psychology* 10 (2016), p. 521. DOI: 10.3389/fnins.2016.00521.

[9] Katharina Gross-Vogt, Hermann Kaindl, and Johannes Dörflinger. “Reflecting on Qualitative and Quantitative Data to Frame Criteria for Effective Sonification Design”. In: *Proceedings of the Audio Mostly Conference*. 2023. DOI: 10.1145/3616195.3616233.

[10] Panagiotis Kantan. “Orchestrating Motion: Real-Time Sonification Tools and Methods to Support Movement Rehabilitation”. PhD thesis. Aalborg University, 2023. URL: https://vbn.aau.dk/files/696001045/_PHD_PRK_ONLINE.pdf.

[11] P. Lopes, A. Liapis, and G. N. Yannakakis. “Sonancia: Sonification of Procedurally Generated Game Levels”. In: *1st Computational Creativity and Games Workshop*. Associated with the International Conference on Computational Creativity (ICCC). 2015.

[12] Alice Peyre et al. “Effects of Different Sonification Types on Upper-Limb Movement in Healthy and Brain-Injured Participants”. In: *Journal of NeuroEngineering and Rehabilitation* 20.1 (2023), p. 78. DOI: 10.1186/s12984-023-01248-y. URL: https://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-023-01248-y.

[13] Ignacio Pimentel-Ponce et al. “Gamification and Neurological Motor Rehabilitation in Physiotherapy Treatments: A Systematic Review”. In: *Physiotherapy Research International* 29.2 (2024), e1974. DOI: 10.1016/j.physio.2023.102197. URL: https://www.sciencedirect.com/science/article/pii/S2173580823000718.

[14] Alfredo Raglio et al. “La riabilitazione neuromotoria con tecniche di “sonification””. In: *Giornale Italiano di Medicina del Lavoro ed Ergonomia* 43.3 Suppl. (2021). Discusses neuromotor rehabilitation using sonification techniques, pp. 42–46. DOI: 10.4081/gimle.549.

[15] Felix Schoeller et al. “Gesture Sonification for Enhancing Agency: An Exploratory Study”. In: *Frontiers in Psychology* 16 (2025), p. 1450365. DOI: 10.3389/fpsyg.2024.1450365.

[16] Giulia Sgubin, Manuela Deodato, and Luigi Murena. “Gamification in Rehabilitation: The Role of Subjective Experience in a Multisensory Learning Context – A Narrative Review”. In: *Gestalt Theory* 45.1-2 (2023). Narrative review on gamification and subjective experience in rehabilitation, pp. 121–138. DOI: 10.2478/gth-2023-0012. URL: https://sciendo.com/article/10.2478/gth-2023-0012.

[17] Martin Starkov et al. “Sonifying motor skills with Pizzicato, a game for motor behavior research”. In: *2024 IEEE Conference on Games (CoG)*. 2024, pp. 1–8. DOI: 10.1109/CoG60054.2024.10645594.

[18] José Ángel Vallejo-Pinto et al. “Applying sonification to improve accessibility of point-and-click computer games for people with limited vision”. In: *Proceedings of the 25th BCS Conference on Human-Computer Interaction*. BCS-HCI ’11. Newcastle-upon-Tyne, United Kingdom: BCS Learning & Development Ltd., 2011, pp. 449–454.

[19] Ziyu Wang, Lejun Min, and Gus Xia. “Whole-Song Hierarchical Generation of Symbolic Music Using Cascaded Diffusion Models”. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2024. URL: https://arxiv.org/abs/2405.09901.

[20] Alexandra M. Zaferiou et al. “A Review of Concurrent Sonified Biofeedback in Balance and Gait Training: Effects and Challenges”. In: *Journal of NeuroEngineering and Rehabilitation* 22.1 (2025), p. 15. DOI: 10.1186/s12984-025-01565-4. URL: https://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-025-01565-4.

[21] Robert J. Zatorre and Valorie N. Salimpoor. “From Perception to Pleasure: Music and Its Neural Substrates”. In: *Proceedings of the National Academy of Sciences* 110.Suppl 2 (2013), pp. 10430–10437. DOI: 10.1073/pnas.1301228110.

9

Evaluation Scripts

This appendix includes all materials used during the evaluation of the end-to-end authoring workflow. It contains the task instructions used for the expert walkthrough, the post-task questionnaire, the semi-structured interview script, and references to the session recordings.

9.1. Participant Information

Nine domain experts participated in the evaluation sessions:

Participant	Background	Role
Tim	Psychology student, will participate in the study with Pizzicato	Walkthrough + Interview
Anouk	Psychology student, will participate in the study with Pizzicato	Walkthrough + Interview
Ivo	Psychology student, will participate in the study with Pizzicato	Walkthrough + Interview
Paula	Psychology student, participated the study with Pizzicato before	Walkthrough + Interview
Noor	Psychology student, participated the study with Pizzicato before	Walkthrough + Interview
Marijn	Psychology student, participated the study with Pizzicato before	Walkthrough + Interview
Defne	Psychology student, participated the study with Pizzicato before	Walkthrough + Interview
Rebeca	Psychologist and researcher	Walkthrough + Interview
Fan Wang	HCI Phd focused in rehibilitation	Walkthrough + Interview

Each expert completed the workflow individually via Microsoft Teams while thinking aloud. Screen and audio recordings were collected with consent.

9.2. Evaluation Procedure

The evaluation consisted of two stages:

- **Stage 1: Expert Walkthrough**. Participants completed representative tasks while verbalizing their reasoning (think-aloud protocol).
- **Stage 2: Semi-Structured Interview**. Experts reflected on usability, flexibility, integration, and applicability in therapeutic and research workflows.

Each session lasted around 60 minutes.

9.3. Walkthrough Task Script

The following tasks were used during the expert walkthrough. They represent the complete workflow required for designing a level for PIZZICATO.

Task A — Music Generation

System link: `http://131.180.119.125:5001/`

1. Select key, tempo, and song structure (Intro/Verse/Chorus).
2. Generate a symbolic piece using the hierarchical diffusion model.
3. Inspect the output and listen to the MIDI.
4. Export the MIDI and its section metadata file.

Measures: Ease of understanding parameters, perceived control, confidence in generating music suitable for therapeutic goals.

Task B — Notes Editing (Trigger--Support Mechanism)

System link: `https://qinxinren1.github.io/Level_Designer/`

1. Load the generated song.
2. Remove sections not relevant for therapy (e.g., the intro).
3. Simplify rhythms using the trigger–support grouping mechanism.
4. Compare normal and reduced versions.
5. Save the edited structure.

Measures: Clarity of piano-roll representation, understanding of note simplification, usefulness for adapting to motor difficulty levels.

Task C — Target Configuration

1. Load the edited musical structure.
2. Inspect note timing using the temporal preview.
3. Assign gestures (manual or auto-assign).
4. Place targets onto the spatial canvas.
5. Adjust difficulty through spacing and density.
6. Export the `LevelConfig.json` file.

Measures: Intuitiveness of gesture mapping, spatial clarity, confidence before exporting.

Task D — Runtime Testing in PIZZICATO

System link: `https://qinxinren1.github.io/Pizzicato_Level_Editor/`

1. Load the exported level configuration.
2. Play the level in PIZZICATO.
3. Inspect rhythm–feedback alignment.
4. Review the CSV performance log.

Measures: Consistency between authored design and runtime behavior, clarity of feedback, usefulness of logs for therapeutic assessment.

9.4. Post-Task Questionnaire

Participants rated each statement on a 5-point Likert scale (1 = strongly disagree, 5 = strongly agree).

Usability

- The interface was easy to understand.
- I felt confident completing the task.

Flexibility

- I was able to adjust the level to different difficulty needs.
- The tool supports designing for different therapeutic goals.

Extensibility / Integration
- The exported level behaved as expected in PIZZICATO.
- The workflow fits real clinical or research processes.

9.5. Semi-Structured Interview Script
The following questions were used for the post-session interviews (15–20 minutes).

Overall Experience
- How would you describe your overall experience using the workflow?

Music Generation
- Was the music generation interface clear and easy to use?
- Which parameters were most or least understandable?

Notes Editing
- How easy was it to simplify or edit notes (trigger–support mechanism)?
- Did the piano-roll representation help you understand the structure?

Target Configuration
- How intuitive was assigning gestures and positioning targets?
- Was the connection between spatial layout and motor difficulty clear?

Preview & Validation
- Did the preview correctly help you anticipate gameplay behavior?

Integration with PIZZICATO
- Did the exported level behave as you expected?
- Was the sonification feedback clear for clinical use?

Therapeutic or Research Usefulness
- How well would this tool fit into your therapeutic or research workflow?
- Which types of patients or experiments would best benefit from it?

Improvements
- What improvements or additions would you like to see?
- Is anything missing that would prevent independent use?

10

Example Level JSON Configuration

This appendix presents a concrete example of a full level configuration exported by the proposed authoring pipeline. The file is a JSON document that encodes both the musical structure (tempo, key, tracks, notes) and the spatial–gestural mapping required by PIZZICATO at runtime.

Structure Overview

At a high level, the JSON file is organised into two main parts:

- `project`: metadata about the project, such as name, version, and creation time.
- `music`: musical and interaction content, including tempo, time signature, key, tracks, notes, and sections.

The `music` object contains:

- `bpm`: tempo in beats per minute (e.g., 90 BPM).
- `timeSignature`: numerator and denominator (e.g., 4/4).
- `key` and `isMajor`: encoded key and modality.
- `tracks`: an array of tracks (e.g., melody and chords). Each track has:
  - `name`, `id`
  - `notes`: an array of note objects.
- `sections`: high-level structural segments (e.g., type A, bars 0–4).

Each note object includes both musical and sonification-relevant fields:

- **Musical fields**:
  - `pitch`: MIDI pitch number (e.g., 83 for B5).
  - `startTime`: onset in beats (relative to the beginning of the piece or section).
  - `duration`: duration in beats.
  - `velocity`: MIDI-like intensity.
  - `noteName`: human-readable pitch label (e.g., `"B5"`).
- **Interaction fields**:
  - `type`: either `"trigger"` or `"support"`. Trigger notes are turned into playable targets; support notes are automatically sonified when a corresponding trigger is hit.
  - `finger`: finger index used for the pinch gesture (1–4), or `null` for support notes.
  - `position`: normalised spatial coordinates on the game canvas:

    - `x`, `y`: values between 0 and 1 encoding horizontal and vertical position.
    - `assigned`: whether this note has been placed on the canvas by the level designer.
  - `supportNotes`: for trigger notes, a list of IDs of associated support notes.
  - `triggerNote`: for support notes, the ID of the trigger note they belong to.

Together, these fields allow the runtime to reconstruct: (i) the musical content, (ii) the timing of targets, and (iii) the spatial–gestural mapping for each note.

JSON Excerpt of a Single Level

This appendix shows an excerpt of a level configuration file exported by the authoring tool. The JSON file contains musical metadata, note events, gesture assignments, and spatial target positions used by PIZZICATO.


```
{
  "project": {
    "name": "generation-0",
    "version": "1.0",
    "created": "2025-11-03T19:24:56.645Z"
  },
  "music": {
    "bpm": 90,
    "timeSignature": { "numerator": 4, "denominator": 4 },
    "key": 2,
    "isMajor": false,

    "tracks": [
      {
        "name": "red_mel",
        "id": "track_red_mel",
        "notes": [
          {
            "id": "note_1762080618237_nutokaye3",
            "pitch": 83,
            "startTime": 0,
            "duration": 2,
            "velocity": 100,
            "type": "trigger",
            "noteName": "B5",
            "finger": 1,
            "position": { "x": 0.4026, "y": 0.3892, "assigned": true },
            "supportNotes": ["note_1762080618237_a9rv2waht"]
          },
          {
            "id": "note_1762080618237_a9rv2waht",
            "pitch": 78,
            "startTime": 2,
            "duration": 2,
            "velocity": 100,
            "type": "support",
            "noteName": "F#5",
            "triggerNote": "note_1762080618237_nutokaye3"
          }
        ]
      },

      {
        "name": "red_chd",
        "id": "track_red_chd",
        "notes": [
```

```
          {
            "id": "note_1762080618237_io70scvh0",
            "pitch": 47,
            "startTime": 0,
            "duration": 2,
            "velocity": 100,
            "type": "trigger",
            "noteName": "B2",
            "finger": 1,
            "position": { "x": 0.3691, "y": 0.2385, "assigned": true }
          }
        ]
      }
    ],

    "sections": [
      { "type": "A", "startBar": 0, "endBar": 4, "noteCount": 41 },
      { "type": "A", "startBar": 4, "endBar": 8, "noteCount": 37 }
    ]
  }
}
```


The full JSON file is stored with the experiment materials and can be provided to the examination committee upon request.