Integrating an Automatic Tonal Expert Agent into a Real-time Music Environment

Jakab Pilaszanovich

Master Thesis UPF / 2013

Master in Sound and Music Computing

Master Thesis Supervisors:

Sergi Jordà

Agustín Martorell

Department of Information and Communication Technologies

Universitat Pompeu Fabra, Barcelona

Universitat Pompeu Fabra Barcelona

Abstract

The idea of connecting the realm of symbolic music making and the direct use of audio is a relevant problem in today's music production technologies. In an effort to overcome these issues two different versions of musical expert agents were implemented for a real-time environment. These served as a basis for providing musical knowledge to the instrument player by suggesting multiple sets of symbolic tonal material. The focus was on the generative methods in which the link between the suggested notes and the simultaneously played sound files was made. After conducting a qualitative user test, the overall observation was that players preferred using the automatic expert agents to the manual scenario. In addition, it was concluded that people with less musical knowledge found the automatic tools more useful than people with more musical background.

Table of Contents

1 Introduction

Current music production technologies provide a platform for creative pursuits in the digital domain for a wide range of users. However, state of the art digital audio workstations (DAW) do not offer a variety of support for creative exploration, neither show a significant tendency towards the stimulation of inspiration. In order to overcome the drawbacks of today's music production technologies, a system of integrated musical expert agents could be introduced in such music making environments.

Nowadays, a large amount of electronic music production and performance is based on the direct use of audio in the means of sound samples, loops and audio files. Inevitably, this resulted in the easily accessible and quickly usable techniques where the producer is not required to have any sort of musical training. The use of non-symbolic musical material clearly democratised digital music making, opening up the opportunity for non-experts to explore their creative potentials. However, the lack of knowledge in traditional music theory can lead to unwanted musical results when combining audio/sound files with symbolic music material, and therefore can have a demotivating effect on those who are not musically trained.

Introducing simulated musical agents can bridge the above-explained gap between non-trained music producers and current music making techniques. These expert agents could provide musically meaningful information to the user, which would result in a musical expert agent system integrated into a creative environment. Such a system could give the opportunity to amateurs to learn and develop their knowledge while creating new musical material, avoiding the technical difficulties and personal limitations they might face. At the same time, producers with higher musical knowledge could benefit from the speed and the ease of creative flow.

The theoretical framework of this project lends itself to the observation that it could provide a platform for merging the research based Music Information Research (MIR) techniques with the user-centred needs of real-time music making systems. On one hand, the academic/theoretical approaches in MIR studies could be easily brought to and tested by a large amount of users. On the other hand, the anthropocentric human-computer interaction (HCI) community would benefit from the applied knowledge of MIR.

Moving from the general concepts and the demonstration of the shortcomings of current music making systems, the problem has to be boiled down to a specific implementation. For the purpose of this master thesis, this implementation has to focus on one aspect of

the musical expert agents. This will be the knowledge of harmony and tonality extracted from the audio files pre-selected by the user.

1.1 Motivation

As mentioned above, contemporary music production systems have some shortcomings. The direct use of audio (loops, samples, recordings, etc.) became a prevalent way of music making. At the same time using MIDI and note-based frameworks provide the fundamental means of musical information in these environments. By the ubiquity of personal computers and the popularity of music making, the usability of both of these music production methods have greatly advanced in the past decades in their respective realms, however there has not been significant research and development in connecting them.

Using audio material in music production in combination with MIDI requires a high level musical knowledge in order to achieve a sufficient musical result. In other words the producer should know and understand the musical features of the audio signal when wanting to combine it with other tone generators, sequencers, arpeggiators, etc. Another aspect worth mentioning is that general digital audio workstations (DAW) as well as lower level programming languages for music production do not tend to offer musical knowledge support. Therefore, one without a certain level of musical knowledge has very limited opportunities for creative exploration.

Most of the above drawbacks become more prominent in the real-time context, when the music production and performance is happening live. A musician using loops/samples in real-time could greatly benefit from a musical expert agent that provides musical information and knowledge in relation to the audio materials being used. Not only providing this knowledge, but having a prepared system that offers a set of choices could give some inspirational, technical and musical support in live performance. This means that the player is not required to have prior advanced musical education and also the computer system could keep the producer in the creative flow of music making.

1.2 Goal

In order to offer a possible solution to the above problems of present music making systems, this master thesis project explores and discusses a potential way of integrating such a musical knowledge system into real-time music performance. The fundamental aim of the project is to develop a basic musical knowledge system that provides musically meaningful information about the sound files being played in real-time. Since the implementation is intended to be used in a live performance context, it is clear since the beginning that the musical information that is being provided “on-the-fly” has to be dynamically changing. This means that as the playback of the sound files are changed while playing, the musical support system has to be modified accordingly in real-time.

The musical information that is made available for the user can take on a wide range in terms of musical features. The following work focuses on tonality and harmony and explores how these can be generated and used to meet the above requirements. The final outcome is an integrated musical agent, which provides a musical help in a tonal context.

To be able to implement and test such a system, the Reactable electronic instrument and music production environment was chosen. The choice was made since it contained all the features that were needed for developing such a system. It is a real-time music production and performance instrument that uses pre-recorded audio material, sound generators, sequencers and global objects that control and filter the MIDI based outgoing signals, therefore allowing the mix of audio with symbolic music material.

2 State of the Art

The introduction of musical expert agents requires the understanding of state of the art research techniques in MIR. The primary focus of MIR is on extracting musical information directly from audio signals, which can be used as the first step towards the ability of the integration of the proposed musical agents. Therefore the following state of the art research will focus on recent studies about MIR integration in real-time music making and performing systems as well as some of the related theories and practices in current musical information extraction in the field of tonalities and harmonies.

2.1 MIR in real-time music applications

The history of real-time (live) music performance and music creation dates back to the years when the possibility of creativity arose in the computer research community. Since the second half of the 20th century, there has been a vast amount of research and experimentation on the possibilities of live music performance, however these systems were limited to the development in research laboratories. The people who have done these experiments were professionals and musical experts before computers became affordable to the general public. The ubiquity of personal computers increased the demand for music making and performing systems, which enhance playability and offer creative opportunities to those who are not necessarily musically trained. Although MIR, as a research field, did not have its roots based on the demand for creativity and to serve live music performance, recently there has been an interest in research to integrate these theories and applications for such purposes. The following paragraphs will discuss some of these implementations.

One of the main topics of recent research in integrating MIR into real-time music making and performance is concatenative synthesis. This technique uses a kind of synthesis where a target sound (or sound file) is assembled by short fragments of audio, which were taken from a larger database [Schwarz, 2005]. In more detail, the database of sound files are segmented into smaller units and then analysed according to the descriptors defined by the algorithm. After the analysis, these segments are selected and ordered in a way to best match the segments of the target sound. Once matched, the segments can be transformed and concatenated, resulting in the synthesis of the original (target) sound. Since, in the past few years technology has reached a point where sample based, high-quality sound synthesis is not of a problem anymore, concatenative synthesisers became a flexible and widespread means of musical creation. Some of the implementations [Bonada and Serra, 2007] incorporate a number of synthesis parameters including the characteristics coming from the gestural control of the instrument. This yielded a research [Maestre, 2009] on the merging of expressivity to

the field of concatenative synthesis, therefore linking the analytical theories of MIR to human interaction with computer music.

Another example of extending the use of concatenative synthesis to the realm of gestural control and re-embodiment of recorded musical material was presented in [Schnell, 2011]. Using a modular wireless motion capture device - the *MO* - the user can control real-time signal processing components with motion. Again, the emphasis of this project is on the combination of MIR and real-time interaction and as a side note, the article refers to the emerging commercial use of interactive games and the popularity of gesture control being embedded in them, linking current research to the need of end users.

A similar concept to concatenative synthesis, however coming from a slightly different angle, musical mosaicing also received some attention in recent research. Reflecting on the tendency of the compositional tools and approaches of modern electronic musicians who are more prone to use sound samples (existing musical material) as the basis of their musical creativity, [Zils and Pachet, 2001] point out the problem of manual data handling in such compositional environments. They state that the problem does not only rely on the fact that composers have to select their samples "by hand", but also on the difficulty of the management of the large data sets they have to choose from. In addition, the library of audio snippets in use are usually not segmented and sorted, which also raises the problem of operation with ease. They propose a mechanism, called *musaicing* (*mu*sical mo*saicing*), which allows constructing a musical sequence by specifying the global properties of that sequence. Then the system chooses and builds the segmented sound fragments automatically from the pre-defined sample library. The methodology of the generation of the target sounds was presented as a constraint problem on both the properties of the individual segments (local) and the properties of the entire sequence (global). Assigning values to, and selecting the right samples optimally satisfy the constraints given by the user. The method of musaicing draws on the psychological phenomena that we can perceive sound on a macro level, from the combination and assembling of the appropriate samples on the micro level. As a compositional tool, the power of *musaicing* and the specific use of MIR relies on the fact that arbitrarily high-level of abstraction of sound properties and descriptions can be used as the means for musical creation.

Still in the domain of concatenative synthesis, [Dubnov et al., 2007] describe the concept of the *Audio Oracle (AO)*. Their new structure aims to assure the continuity between the sliced audio segments by indexing them in terms of the detection of approximately repeating sub-clips with variable length, which they call *audio factors*. The name comes from the extension of the *factor oracle*, which is a finite state automaton constructed in linear time and space in an incremental fashion. As they state,

the complexity of the AO allows the algorithm to work in real-time interactive environments.

Another large field of studies cover the concepts of automatic improvisation. The OMax Brothers project [Assayag et al., 2006] introduces a human-machine interaction system in which the computer learns from the performer in real-time. The system was built to be improvisation oriented and its multi-agent architecture uses statistical learning and sequence modelling to achieve automatic accompaniment. These techniques result in a virtual musical partner, which gains all its knowledge based on the musical input of the performer, in a non-supervised way. The algorithm can be divided into two parts. One is directed towards the learning mechanism, which builds the statistical model from the input sound samples. The other one is in charge of the generative part, where a prediction-based sequence is played back following the previously learnt model.

The formal definitions of musical anticipation and anticipatory design were discussed recently by [Cont, 2008a]. These serve as principles in the process and design of applications in MIR and computer music in general. In order to gain access to musical structures and answer the question of "What to expect?", mathematical and statistical frameworks were introduced. The second step described in his thesis deals with the decision making process where the machine learning algorithm uses the previously recorded anticipatory profiles of actions. In the last step the problems of synchronising the performer's actions and the generated material are addressed, where Antescofo [Cont, 2008b], a score following system is introduced.

The possible use cases of machine-learning algorithms in real-time music making and performance environments were also examined by [Fiebrink, 2011]. In her thesis, she presents a work, which discusses the potential application of supervised learning algorithms in combination with both computer music composition and live performance. Her fundamental concept is based on observing and to deeply understand the needs of the user when it comes to using machine learning algorithms. Therefore, according to her, designing a supervised learning application, which is easily usable by musicians, can result in a tool for creative purposes. Along with her thesis, she presents a freely available, open source software project, the Wekinator. This program is designed in a way that throughout the whole supervised learning process it allows human interaction. It is important to observe that the approach taken in this thesis focuses not only on the one-way control of human-computer interaction, but also on the nature of computer-human feedback. A point worth mentioning about the structure of her thesis is that there has been user studies conducted on groups with different musical backgrounds. From the user studies, a more comprehensive description of the priorities and goals of the composers for interacting with machines in instrument design and music composition were obtained.

2.2 MIR in commercial DAWs

As mentioned before, current state of the art DAWs rarely have MIR based knowledge embedded in them. However, some of the most recent versions of popular music production systems do contain certain applications, which are based on extracting musical knowledge from audio files. Although, these applications are at their infancy, it can be seen that there is a growing tendency in the commercial market towards integrating MIR knowledge to music making softwares.

One of the leading loop-based music sequencer softwares in the market is Ableton Live, precisely designed for live performance. The earlier mentioned techniques of audio sample based music production are in the core of this system. In their newest release in March 2013, some new features were added which stem from the integration of musical knowledge extraction. From the tonal aspect two of the features convert harmony and melody from an audio input into MIDI representation. The audio input can be a pre-existing file from the sample library of the user as well as a live performance input (singing or instruments) [Ableton, 2013]. These techniques are along the vein of integrating Melodyne type music analysis features into a real-time context.

Another prominent music production software, Cubase has already included the option of getting tonal information from sound files. The features offered by VariAudio are considered as the headline tools of Cubase 5. The most obvious application is the pitch correction of instrumental solos and vocal inputs. Another feature is the ability to segment audio files into musically meaningful parts, mainly splitting the melody line into different parts according to the separate notes. Although the VariAudio extension was originally made for correctional purposes in the music production phase, it also offers some creative guidelines on how to arrange melodic lines and generate harmonies. In their latest release (version 7), they introduced a global chord-tracking feature, which lets the user see and follow the estimated scale, chord, harmony and key structure detected for both audio and MIDI material. Using this feature, all other tracks in the working environment can be set to automatically adjust to the harmony of the given musical phrase. In addition to that, a “Chord Assistant” allows the user to create harmonic chord progressions by the algorithm suggesting the next chord event, taking into account the previous one [Cubase, 2012]. The user can control the complexity of the harmonic rules of the generated sequence. This results in a music recommendation system in the creative stage of music production, where the user is not required to have a deep knowledge of music theory.

2.3 MIR in harmony and tonality extraction

Using computational models to extract information from audio signals has been the primary focus of the MIR community. The types of information that have been retrieved

vary on a wide range, however the approaches and theories can usually be categorised into two parts: context and content based. For the purposes of this thesis, the review of the theory and applications will concentrate on the content-based approach, in which knowledge about a signal is acquired by direct analysis, using the recorded sound itself. In this case the signal inherently carries all the information needed. The following paragraphs are a short overview of the interests in audio signal analysis, which concentrate on the ability to retrieve tonal and harmonic information from audio.

2.4 Tonality induction

In music psychology and the cognitive sciences there has been much effort put in the study of the cognition of tonality in Western music [Longuet-Higgins and Steedman, 1971; Temperley, 1999; Krumhansl, 2000; Chew, 2000]. An early theory and representation of tones and their relationships was proposed by [Shepard, 1982] where the close octave relationship between pitches is demonstrated on a spiral. The set of pitches, which are an integer number of octaves apart from each other, are called pitch-classes. The pitch quality of tones is also referred to as "chroma" therefore a pitch-class can also be defined as the set of all pitches sharing the same chroma.

Psychological experiments were carried out by Krumhansl and collaborators [Krumhansl, 1990] to determine the degree of perception of individual musical tones within a tonal context. This resulted in a quantitative measure of the hierarchical ordering of tones related to a given tonal centre or a given key. In terms of music theory, this is a representation of the structural significance or relative stability of the functions of tones within a given tonal context. The following figure demonstrates the result of the measurements. It can be seen that the graph is split into 12 histograms, where each represents a pitch-class in the Western musical scale. The height of the histograms corresponds to the relative "strength" of the tones. As it can be seen, the tonic received the highest rating, followed by other tones part of the scale, and the non-scale tones having the lowest rankings in both the major (top) and minor (bottom) contexts.

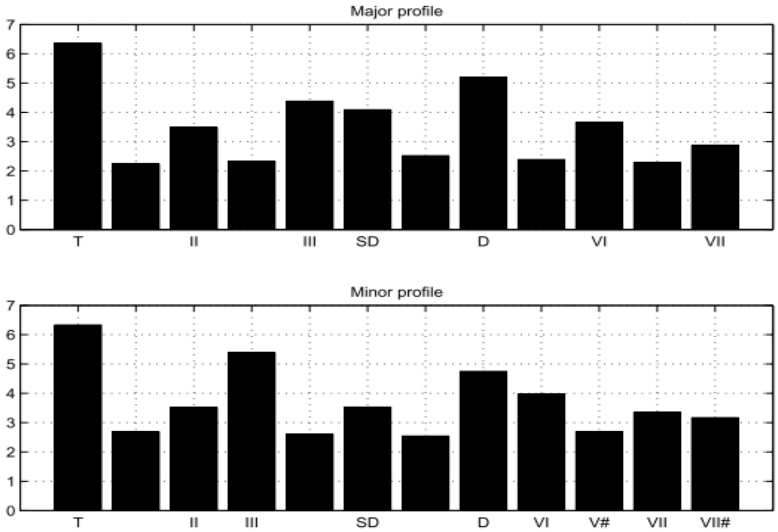

Figure 1. Major (top) and minor (bottom) pitch-class profiles as proposed by Krumhansl and Schmuckler [Krumhansl, 1990].

After conducting the probe-tone experiment on human subjects, they proposed a key-estimation algorithm based on the analysis of MIDI input files. The key of the given melody was estimated by a set of note duration values for the 12 pitch classes that have appeared in the musical sequence. Comparing the retrieved data to the above described pitch-class profiles one could identify the key/tonality of the melody. This approach of tonality induction is called the *weighted key profile* model.

The implementation for tonality extraction in most cases is based on either Pitch-Class profiles [Fujishima, 1999] or Constant Q-profiles [Purwins et al., 2000]. Fujishima, who has developed a real-time chord recognition system in sound signals, conducted one of the early works on the topic. He describes the task as a process of identifying simultaneous pitches, which make up the harmonic palette of traditional Western music. The final result of his work is to infer the chord name and its type, however in our case the importance has to be shifted towards the actual method, which was used to acquire such information. The main approach was to analyse the input sound in the frequency domain and transcribe the polyphonic signal in order to be able to identify the individual musical notes. These results could then be turned into musical symbolic representation and therefore the Pitch-Class profiles (PCP) were obtained. These PCPs then show how much of a tone is relevant within a given piece of musical signal.

Another method for describing tone relevance in audio signal is the use of constant Q-profiles (cq-profiles), mentioned earlier [Purwins et al., 2000]. The goal of their work was to approximately represent the tonal centre of a given input sound. Similarly to the above method used by Fujishima, cq-profiles are 12 dimensional vectors in which each component corresponds to a well-tempered pitch class on the chromatic scale. They can

be viewed as a new approach to represent key profiles. The basis of the model is the calculation of the Constant-Q Transform, which is a filter bank where the ratio of the centre frequency and the bandwidth (Q) is constant, and the centre frequencies are geometrically spaced along the spectrum. A cq reference set is built from pre-analysed small pieces of music or from sampled cadential chord progressions. These sets give rise to a sequence of 24 cq profiles corresponding to all the keys in major and minor. These sets serve as the basis when comparing the obtained cq vectors from the audio files from which the tonality of the given piece can be obtained. As they discuss in their paper, the cq profile method is a powerful tool and they state that it can be extended to a real-time context.

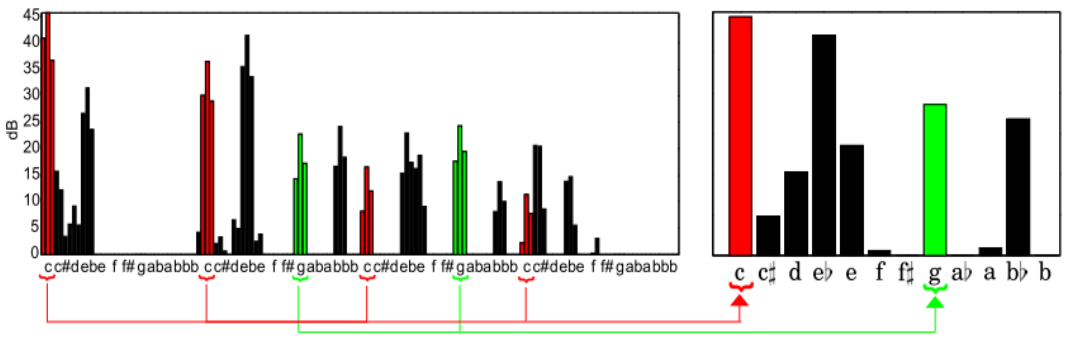

Figure 2. The constant-Q transform of a minor third, c-e played on a piano with 3 bins per half-tone (left). The constant-Q profile (right) is obtained by summing up the bins for all the tones over all octaves.

A further work about the description of tonal content in musical audio signals was carried out by [Gomez, 2006]. One of the main focuses of this thesis is the presentation of an algorithm, which computes the descriptors necessary for pitch class distributions. It also entails and accepts the idea that depending on the application context, there is no need for a perfectly accurate symbolic transcription of audio data. Therefore it encourages the realisation of such applications where the gap between symbolic-oriented methods and audio streams can be bridged. The outcome of the thesis can be thought of as an extension of the pitch class profile (PCP) method presented in [Fujishima, 1999], resulting in an approach called *harmonic pitch-class profiles* (HPCP). This additional feature takes care of the nature of the acoustic behaviour of musical sounds, where it is known that the sound of an instrument (or voice) contains the upper harmonics of the fundamental frequency. Taking this into consideration, the HPCP method can build a more advanced and accurate representation of the pitch-class profiles. From the evaluation of the HPCP approach, it was concluded that it is robust against noise and is not dependent on the timbre of the instrument. In addition, it works for both monophonic and polyphonic signals. The final product of the thesis is called the HPCP - Harmonic Pitch Class Profile vamp plug-in. It was designed to extract

features from audio, which allow for computation of the instantaneous evolution of HPCP in the given signal.

It is rare to find MIR research, which takes into account the fact that key and chord description of audio signals is inherently ambiguous. The problem of finding a ground truth in such areas is not only a problem that technologists have to face, but its ambiguity is also a fundamental question in the cognitive sciences. In order to avoid the problems coming from the comparison of retrieved tonal information to a given categorical ground truth, this project will only focus on the real appearance and relative significance of tones in an audio signal.

2.5 Summary of the state of the art

To summarise the previous chapters, the topics of the use of MIR under the light of human-computer interaction has been discussed through the overview of some of the major projects and research directions in the field. Most of them can be used as a basis for this master thesis project on more of a conceptual level, however some of the more practical approaches can be borrowed as well. The above explained applications and the theories related to them are the fundamentals to a project, which is a preliminary implementation of a creativity support system focusing on tonality.

3 Methodology

3.1 Reactable

As mentioned above, the Reactable is a real-time music production and performance environment. Below a schematic diagram shows the mechanism and the flow of data that has been used for the project.

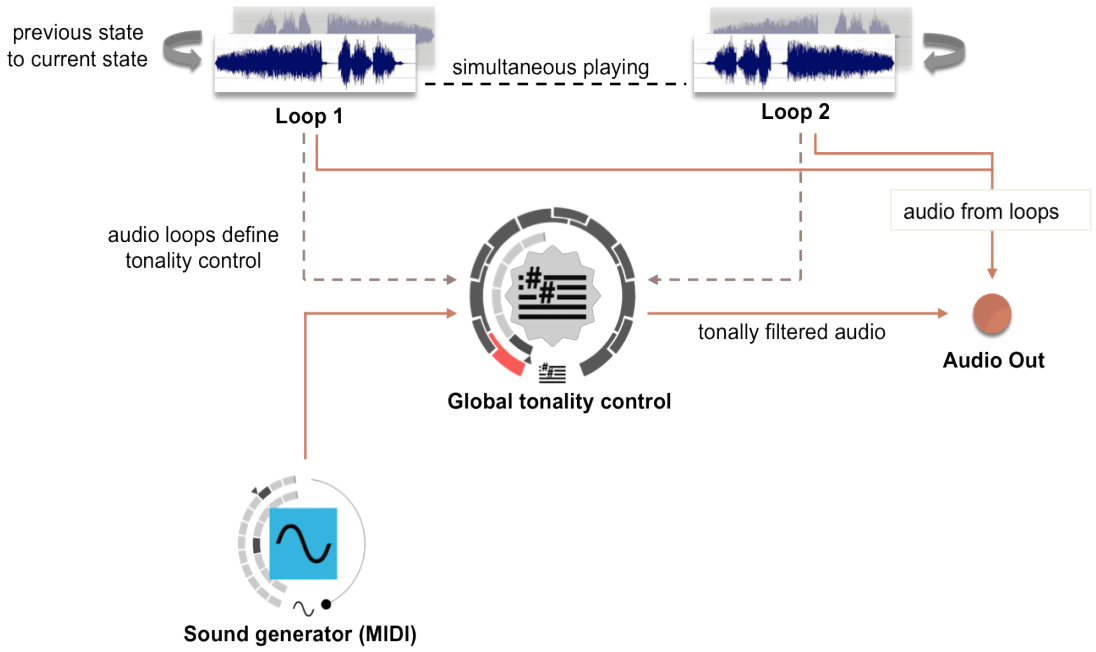

Figure 3. Schematic diagram of the working mechanism of the Reactable used in the project

The two highlighted audio images on the top of the diagram represent two audio loops that are being played on the Reactable. As noted, they are played simultaneously. The dashed lines coming out from the two loop players directed towards the *global tonality control* (tonaliser) object indicate that the tonality is defined by the musical content of the audio materials themselves. While providing the musical information to the tonliser, they are sending the audio to the output. The object positioned in the bottom left of the diagram is a sound generator. It generates MIDI based notes; therefore the tones that it produces are part of the 12 note chromatic musical scale. The arrow pointing from the sound generator to the tonaliser demonstrates that the incoming MIDI notes get filtered, and only the ones that were defined by the audio loops shall pass towards the audio output. One last aspect of the working mechanism, which is hinted in the top left corner of the diagram is the importance of the previous loops that have been played prior to the current one. To be able to offer a musically meaningful set of notes to the tonaliser object, incorporating the "history" of the musical evolution was an essential point.

The above diagram gives the initial framework to the project and defines the fundamental features and requirements for the application. After setting up these initial aims, the project's main focus was delving further into the process of generating

musically meaningful tonal information for the tonaliser based on the audio files that are and have been playing. The two main concerns were dealing with the generation of tonal material when audio files are simultaneously playing; and the question of how to incorporate the tonal state of the previously played samples.

3.2 Global tonality control object - Tonaliser

The tonaliser is a global tonality control object on the Reactable. It means that once it is put in the Reactable environment, it controls all sound generators, sequencers and arpeggiators (i.e.: all MIDI based objects) that are being played.

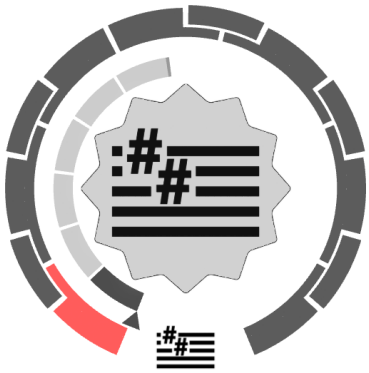

Figure 4. Image of the tonaliser object (as it appears on the Reactable interface)

The way it controls these objects is basically a filter mechanism of incoming MIDI messages. It has 12 “switches” which correspond to the 12 notes of the chromatic scale. Each switch can have a binary value attached to it: either on or off. Only those notes that are switched on will be sounding in the audio output.

As discussed above, the idea of the thesis project is to implement an algorithm that controls these switches in a musically meaningful way. As the user changes the audio loops, their tonal content would drive which MIDI notes should be filtered generated by the other Reactable objects. This would lead to a dynamically changing tonal support system.

Another important feature of the tonaliser is that it has 6 presets, which can be switched in between by the user in real-time. These presets can offer different sets of notes (chords, scales, note combinations) to the user while producing or performing. Following the arguments from above, this feature of the tonaliser comes in very handy for the idea of having a type of recommendation system where these presets are filled up with different combinations of notes that would all match with the audio files that are being played.

4 Implementation Structure

The above-explained layout of a working mechanism suggests that the implementation can be divided into two main parts. The first part is the *analysis* of the audio files. This means that the appropriate musical information (key and scales) is extracted from these files before the application is running. For this reason we will refer to it as the "off-line" part of the project. The second part is the *generative* part, where the algorithm takes care of providing the musically meaningful information to the user in real-time. Therefore this process will be referred to as "on-line".

This approach lends itself to some limitations. In the first place, the audio files should be well suited for the purpose of the project in order to be able to test the outcome. These prerequisites primarily include the fact that they should carry tonal information as opposed to non-tonal (e.g.: they should not be percussive sounds exclusively). Another aspect is the choice of these audio files in terms of their relation to each other in the realm of tonality. For the purpose of the project and the possibility for straightforward testing, they should be tonally well matching loops. Having said that, they are certainly not restricted to be in exactly the same key, however there should be some relatively close tonal relationship between them.

4.1 Tonal analysis

The analysis part of the project comprised of extracting tonal features of the prepared sound files. Initially, Essentia, an audio analysis tool developed at the Music Technology Group (MTG) was used for the tonal analysis. It offered a variety of music related features, both on low and high level of abstraction. The ones that were found to be useful for the purpose of this thesis project were related to key, chord and scale characteristics of the files. These features provided values on the global scale of the sound files, meaning that the evaluation was done on the audio as a whole. In addition to the global features, there was an option to observe the "chord progression" within the sound files on a frame-by-frame basis, therefore time information was included in the analysis.

Although Essentia offered the above features, which suffice for a direct tonal analysis of the files, the extracted musical characteristics were required to offer a more elaborate version of tonal analysis for this project. The reason for the need of a more complex outcome from the analysis part relates back to the previously discussed features of the project's mechanism. As the tonaliser object on the Reactable offers the option of having presets, there was a need for the analysis part to provide a set of tonal possibilities, rather than only one global feature. Therefore, the decision was made to focus on a tonal analysis method that results in a multiple number of candidates for extracting keys.

The following diagram shows the logical flow of the analysis of the audio files.

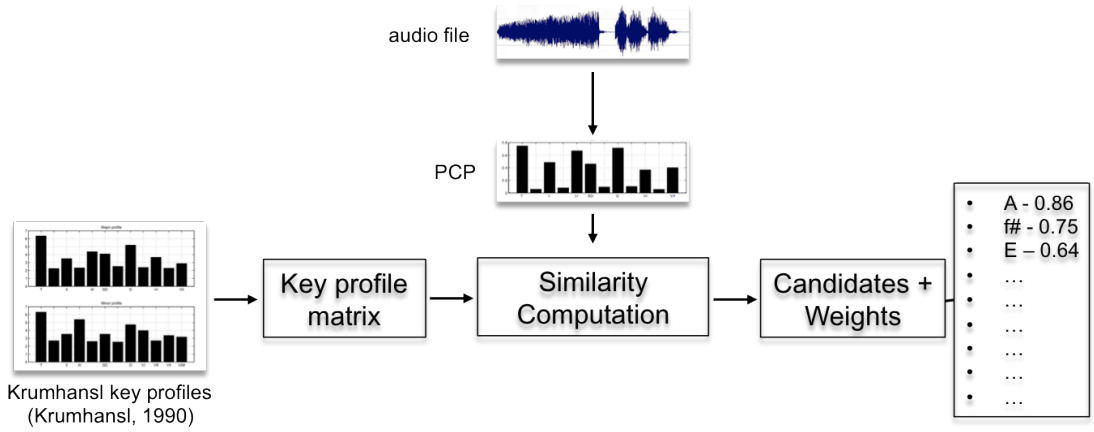

Figure 5. Key extraction mechanism

The left hand side of the diagram shows the Krumhansl key profiles (Krumhansl, 1990), which were discussed in the state of the art section. These key profiles serve as the basis for building a key profile matrix, which is essentially the extension of the Krumhansl profiles to all the possible major and minor keys of the Western scale (12 major, 12 minor). These prepared key profiles then provide the references for the key computation of the given audio files. In order to be able to compute the similarity to these key profiles, the pitch class profiles of the audio files have to be extracted (upper part of diagram). After extracting the PCPs, the similarity computation compares these 12 dimensional vectors to the Krumhansl profiles. The result gives 24 normalised values for each possible key (12 major, 12 minor). These represent the "closeness" of the audio's tonal content to the predefined key profiles. After reordering the results from the similarity computation, one could achieve a hierarchy of possible key candidates with weights attached to them (right hand side of the diagram above).

This *weighted key profile model* for obtaining tonal information from the audio files met the goals of having a set of key candidates. In addition to having the key candidates, the values of the correlation strengths (weights) are also attached them. This gives the possibility of having a quantitative measure of "how much" a given key is present in the audio file.

4.2 Strategies for generating tonal material

After having extracted the tonal information from the sound files, the work had to concentrate on the main questions of the thesis: the investigation of a possible algorithm that generates the tonal material in real-time. The central concept of the outcome of the generative part was to attach musically meaningful information to the sound files. The core implementation of the project explores this idea of "musical meaningfulness" and what alternatives can be offered to achieve sufficient musical results.

Extracting higher level concepts like key, harmony or scale from the signal immediately creates an abstracted version of representation. Although using this symbolic level system adds musical meaning and context, it comes with the fact that some information contained in the signal might be lost in the process of abstraction. As an example, an audio file might contain a richer tonal structure than the symbolic level could represent, especially when the outcome is confined to the standard chromatic major and minor contexts. On the other hand, the lower level PCP representation shows information that is "closer" to the original signal, however without including any musical theory.

Aside from these observations about the outcome and features of the analysis part, the project had to focus on the ability to integrate them with the application goals. As implied on Figure 3 (working mechanism diagram), the two main concerns that should have an effect on the tone generation are the *simultaneous* playing of multiple audio files and the relationship between their previous and current states – *successive* playing.

The importance of storing and using musical information from previously played audio samples has a music theoretical answer. The evolution of a musical phrase, the progression of harmony and sequences of chords/scales are all relative movements within the tonal structure of music. Therefore, the idea of using the short term "history" of musical evolution can enlarge the palette of musical knowledge in a given moment. This idea can be introduced to the implementation by adding state machines. These state machines have the role of storing the previous state of a given system and perform calculations under the light of the stored state. This way the previous state can have an effect on the choice of the current one.

The fact that the outcome of the analysis part of the project offers a set of key candidates drives the approach of the state machine's working mechanism. In a system like this, the state machine can serve as the agent that makes a decision on which candidate to use. The choice has to be dependent on the current state's candidates (and their weights) and the previous state that the system was running at. In more detail, the state machine should favour those current candidates that are more similar to the previous state. In order to be able to implement such a state machine, a careful definition of similarity between states has to be developed, which will be done in the following sections.

To have simultaneous and successive musical information and the observation regarding the low and high level representations (PCPs and symbolic respectively) of audio signals lead to the conclusion that there could be two possible ways of generating tonal material.

The first option is to base the computations of simultaneous and successive playing on the low level characteristics of the signal. This means that the extracted PCPs serve as the main source of musical information. This approach is more mathematical and physical, since - as discussed earlier - it resembles the original audio signal in a more direct way. Therefore this version will be referred to as *signal level approach*.

The second option is to use and compute information on the symbolic level. In this case, the tonal information of the audio files being played at the same time and the previously played ones are represented by the extracted keys. The mixing and combination of these keys is the process in which the tonal template of the tonaliser is generated. This version will be called the *symbolic level approach*.

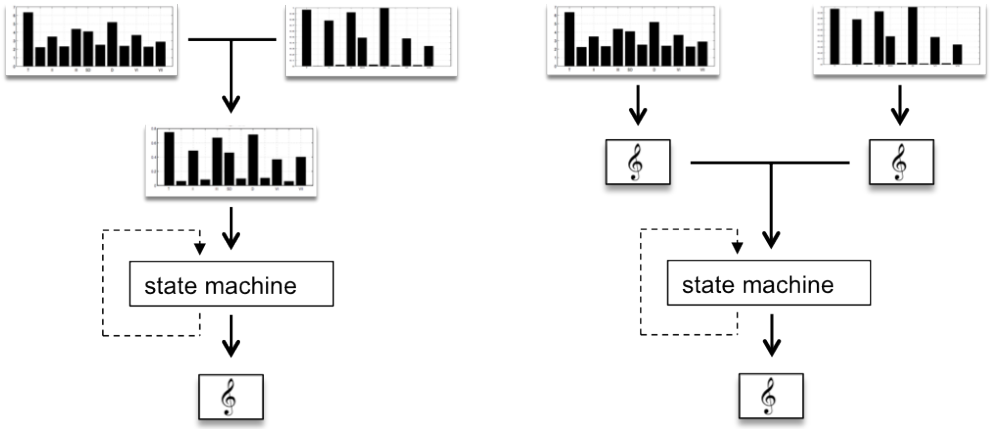

Figure 6. Basic schemas for signal and symbolic level implementation

As discussed earlier, the extracted tonal information offered a set of key candidates. The tone generation methods involved using the corresponding scales assuming the audio files were in the tonic of a given key. Therefore, for example a key in C major would indicate the notes of the C major scale (C, D, E, F, G, A, B). The choice of the 7 note scales was made in favour of other note combinations (e.g.: chords) since they provide a more contextual musical information and a wider range of note candidates including fundamental and passing notes.

4.3 Signal level approach

This version of the tone generation uses the PCPs extracted from the audio files as the basis for computation. The first part of this chapter discusses how musical information is generated when multiple sound files are being played. The second part addresses the problems of similarity between different states in time and how that can be incorporated in the algorithm.

4.3.1 Simultaneous playing

The diagram below shows the schematic of the logic flow of generating the tonal template when sound files are playing at the same time. This diagram does not include the computation regarding the previous states of the audio files, i.e. no state machine is shown.

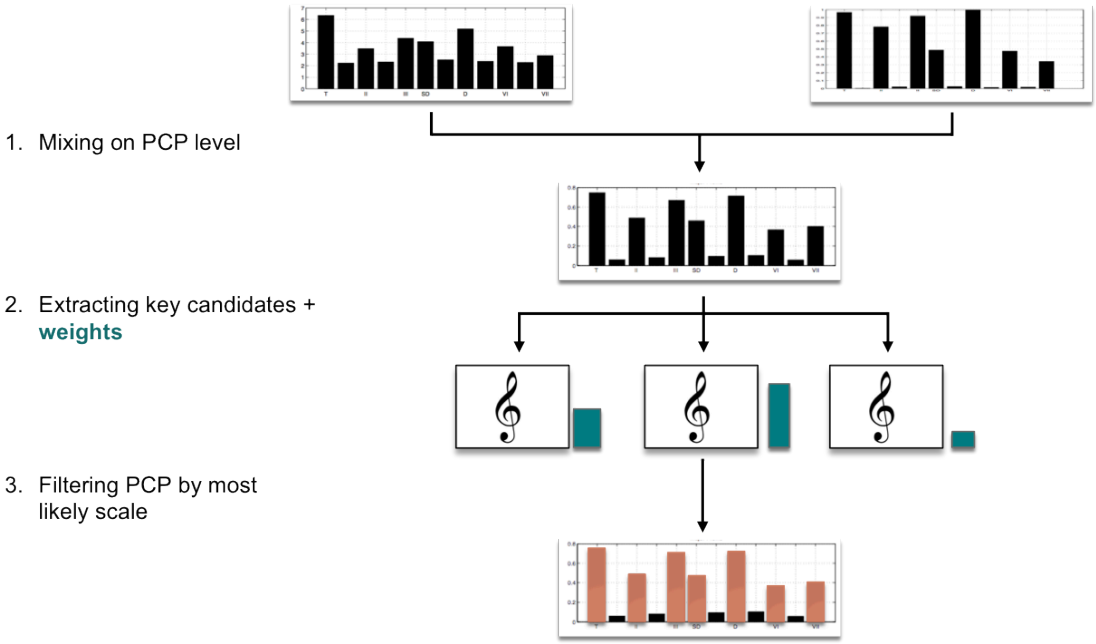

Figure 7. Signal level implementation schematic diagram

The top two PCPs represent the audio files that are being played simultaneously. The way of mixing the PCPs was decided to be adding the two profiles in comparison to multiplying them. The reason for this choice was to retain as much of the tonal information in the signals as possible. In this case addition can be thought of as the union of the two signals, whereas multiplication would correspond to the intersection. After that, the candidates and their attached weights were extracted from the mixed profile. The initial implementation, which does not take into account the previous state, picked the key candidate that has the highest probability. The next step was to filter the original PCP with the notes of the scale corresponding to the most likely key. The filtering is done by multiplying the original PCP with the Krumhansl profile of the candidate. This results in set of 7 notes of a given scale that can be used to play along with the two audio files.

4.3.2 Successive playing

As discussed briefly earlier, including musical information from previous states shall enrich the subtlety of the implementation. For the state machine to have the ability to favour those current states that are similar to the previous one, some sort of similarity or "closeness" of different states shall be implemented. The idea of similarity should

follow the same mind-set in terms of the low level computational approach. Therefore the closeness of the states should be defined on the PCP level.

As a PCP can be thought of as a 12 dimensional vector, the concept of similarity can be turned into a mathematical problem. The similarity of two vectors can be measured by taking their dot product. In other terms it means that the overlap of the two vectors can be quantified by a scalar. If the two vectors are normalised, the result of the dot product yields a number between 0 and 1, the latter meaning perfect similarity.

Following this mathematical/physical approach to similarity, the state machine in the signal level implementation performs the calculation of the dot product between the previous state's PCP and the current state's PCP candidates. These candidates of the current state are given by multiplying the original PCP with the key candidate's corresponding Krumhansl profile. In addition to taking the dot product, the result is multiplied by the weight attached to the given candidate. This is done in order to incorporate the likelihood of the current states as well.

The following diagram shows the signal level application of the tonal template generation including the state machine, thus incorporating the idea of musical evolution.

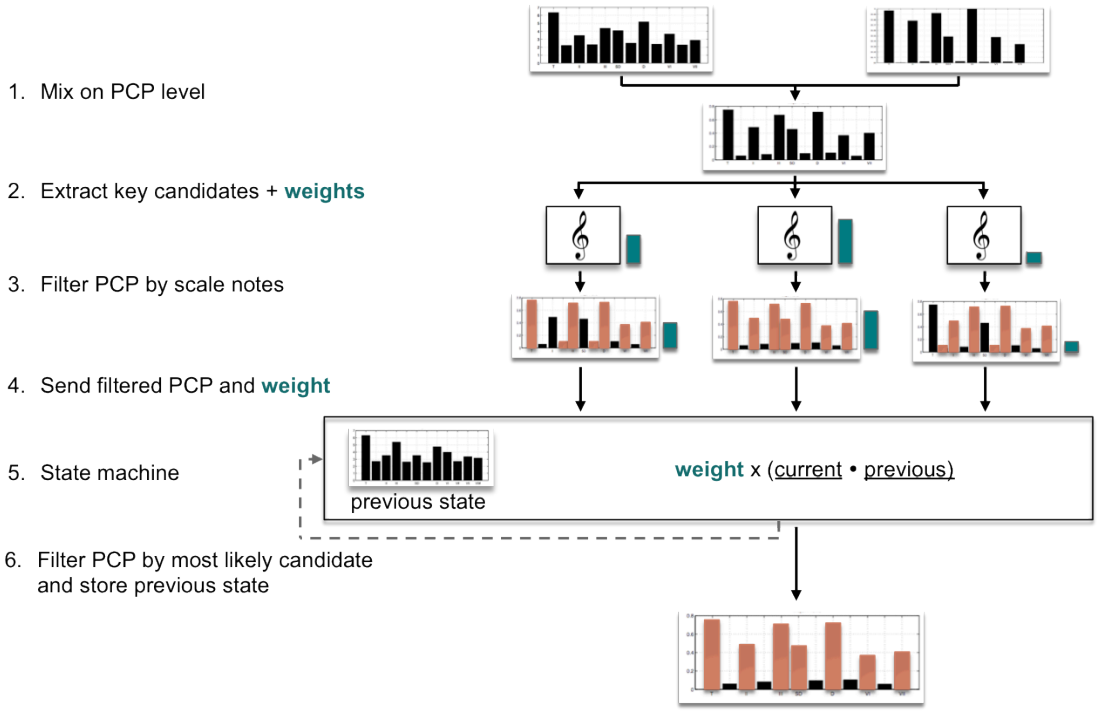

Figure 8. Signal level implementation with state machine

The diagram is very similar to the previous schematic (Figure 7) in its structure, however the state machine is added to the logic flow. The state machine calculations are run before sending the output. Once the current candidates are compared to the previous state, the most likely candidate passes to the output, as well as gets stored in the state machine's previous state, waiting for the next set of inputs.

4.4 Symbolic level approach

This is the second version of the tone generation process of the thesis. In this case higher level of musical characteristics of the audio signals are used. This approach has a more music theoretical basis than the previous one, thus the concepts of simultaneous mixing and successive musical information follow these lines of thought.

4.4.1 Simultaneous playing

As opposed to the method in the signal level implementation, in this version the problem of obtaining musical information from multiple sound sources happens on the symbolic level. The symbolic information retrieval is done for the individual sound files instead of a mix of them. The mixing process is done after having the key candidates and their weights extracted.

In this case the method to combine the two sources is first to multiply the weights of the 3 most likely candidates and at the same time take the intersection of the notes of their corresponding scales. Since 3 candidates with the highest probabilities were chosen from each file, this results in 9 sets of notes with a new combined weight value attached to each of them (see following figure).

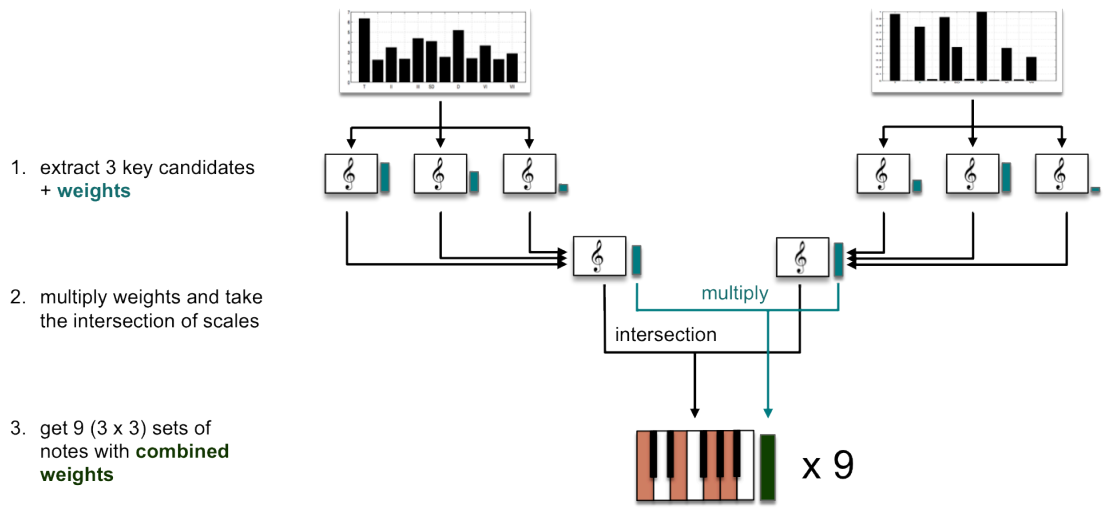

Figure 9. Symbolic level implementation schematic diagram

4.4.2 Successive playing

Similarly to the previous version, the previous state effects the current state by using a state machine. In this case, however there are two state machines that have to be used, since the sound files generate the states separately.

Since this version has an approach on the symbolic level, the similarity between states have to be defined in a more music theoretical perspective than the low level PCPs. A basic idea about measuring musical closeness/relatedness in terms of tonality is the distance on the circle of 5ths. The circle of 5ths can be thought of as a tool that shows how many notes are shared between two scales. Although it is a very simplified problem, but the assumption that the more notes are shared between two scales, the closer they are musically is sufficient for the scope of this project. Therefore, we can say that the distance on the circle of 5ths measures similarity between two successive states in this implementation. Again, the outcome is normalised; 1 corresponds to complete similarity, and zero corresponds to the furthest distance on the circle.

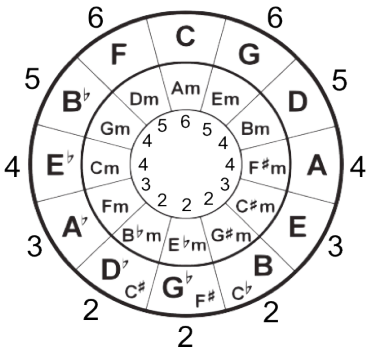

Figure 10. Circle of 5ths and the number of notes shared related to the C major scale

The following diagram is the extension of the previous logic flow (Figure 9). It can be seen that the current candidates are sent to the state machine in each case. The computation in the state machine generates a new set of weights for each candidate. These weights comprise of the incoming weight values multiplied by the distance between the current key candidate and the previously used key. The 9 possible sets of notes are generated in the same way as explained in the previous version, however in this case the modified weights are used. In addition to the previous diagram, the current state that is being used is stored in the state machine's buffer, waiting for the next set of input candidates generated by the new incoming audio files.

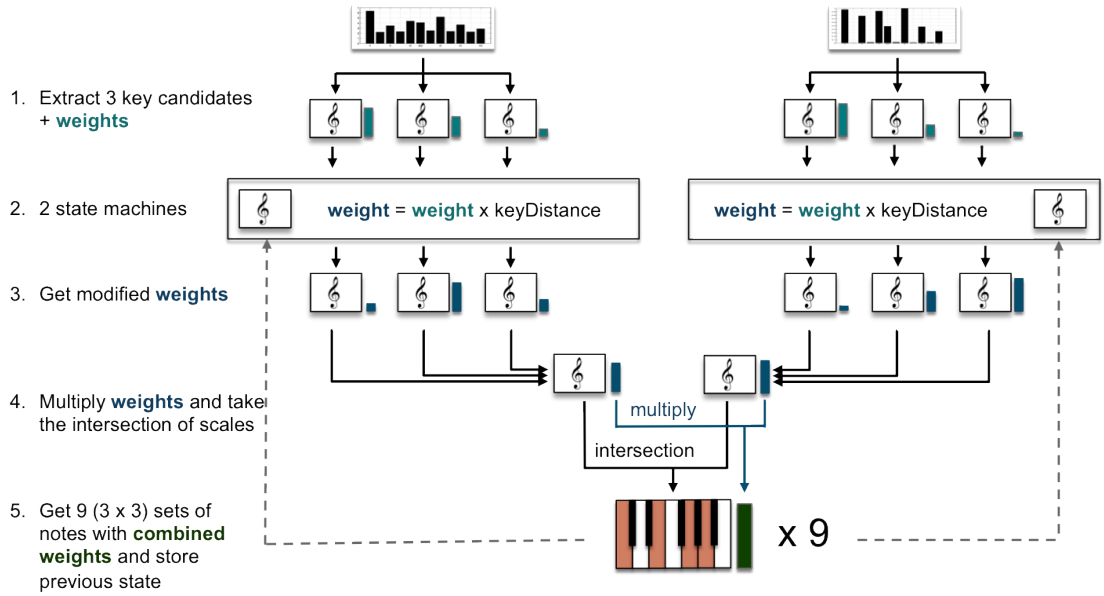

Figure 11. Symbolic level implementation with state machine

4.5 Strategies to fill up the tonaliser

As a final task in terms of the implementation, the previous theories about tone generation in the signal and symbolic level contexts have to be linked to the application of the tonaliser object on the Reactable. The previously explained presets of the tonaliser have to be filled up with the set of possible tonal templates that have been generated by the different methods.

In the case of the signal level implementation, the presets are defined by introducing the highest peaks of the resulting PCPs that have been filtered by the final candidate. This means that the first preset has the 2 highest peaks, the second preset the 3 highest, and so on, until the last (sixth) preset provides the full 7-note scale. This results in an interface, which builds up the notes of a scale, therefore gives predictable and stable suggestions to the user in terms of tonality.

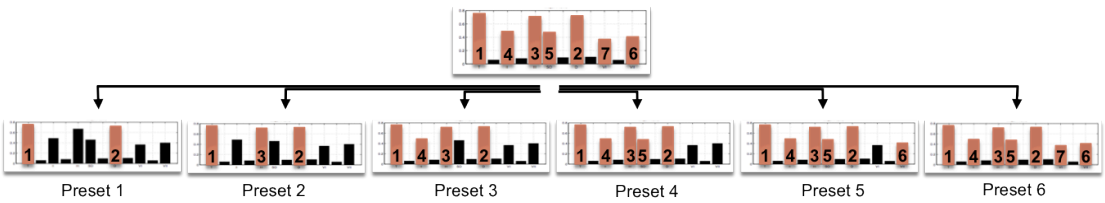

Figure 12. Tonaliser presets for the signal level application

In the case of the symbolic level implementation, the presets are defined by building a hierarchy of the 9 sets of note combinations based on the weight that has been attached

to them. Taking the 6 highest weighted options in a descending order fills up the 6 presets on the tonaliser. Compared to the previous method, this gives a set of presets that are not necessarily musically related, however are more likely to break the boundaries of traditional music theory. Although the offered sets could seem more unpredictable to the user, the musical result could be in interest of more experienced users.

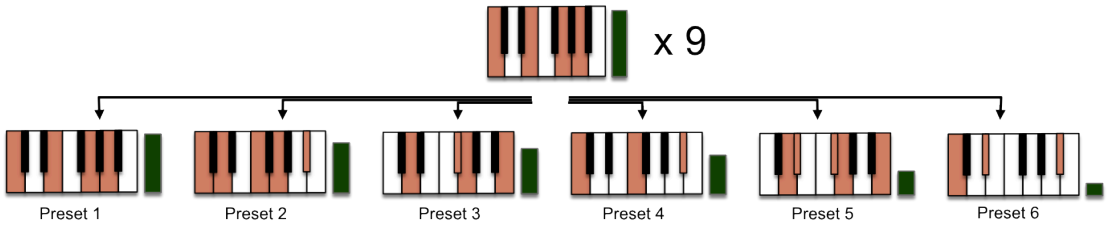

Figure 13. Tonaliser presets for the symbolic level application

5 User Tests

In order to assess the outcome of the algorithms, user tests had to be conducted. The nature of the project lends itself to a qualitative testing scheme, where users have to give their opinion about the usability, performance and enjoyment while using the features of the automatic tonality object. A 5-point Likert scale was used for all the corresponding questions to gather data on such subjective measures.

5.1 Experimental design

The fundamental purpose of the measurements was to see if the use of the automatic tonaliser in its different modes is favoured compared to the manual version. A step beyond this question was to study the preference between the two versions of the automatic tonaliser algorithms.

In addition to the examination of the different modes, another aspect of the tests was taken into account. The initial hypothesis was that people with less musical knowledge would favour the automatic tonaliser compared to the manual one. This assumption was based on the fact that the automatic tonaliser serves as an expert musical agent, providing musical knowledge in real-time. Therefore, one without musical knowledge would benefit from using it. On the other hand, users with sufficient musical knowledge would find the automatic tonaliser as an agent, which confines them to given notes/scales, not leaving an option for individual, free experimentation and control. All in all, the hypothesis needed a testing scheme that relates the preference of the automatic tonaliser to the manual one as a function of musical knowledge.

Before doing the tests, the users were asked to fill out a *pre-test questionnaire*, which measured their own judgement of their musical knowledge on a 5-point Likert scale (see Appendix A). These questions referred to general music knowledge, music theoretical knowledge, ability to play an instrument and compositional experience. After filling out the initial questionnaire, the users were asked to play on the Reactable, testing both the manual and the automatic versions of the tonaliser object in different scenarios.

To avoid the possibility of any bias in the testing, the Reactable had to be used in a reduced setting. This included

- *two loop player objects:* these stored and played the audio files letting the user switch between them in real-time
- *sampler object*: sound generator that played electronic piano sounds
- *melody object*: this object was connected to the sampler object and was set to be working as an arpeggiator, generating a ramp of all the notes of the 12 note scale
- *tonality object*: the filter for the generated notes by the arpeggiator

The following three scenarios were presented to and tested by the users:

1. Manual tonaliser: the tonality object works on a manual basis, where the user chooses the notes that can pass generated by the arpeggiator. The volunteers are free to define their own presets while playing
2. Automatic tonaliser 1: the tonality object works on the basis of the first implementation, where the presets are filled up automatically according to the given scale
3. Automatic tonaliser 2: similarly to the previous scenario, the tonality object is defined automatically, but here the mechanism follows the second algorithm

In addition to the simplified setting of the Reactable, by showing an explanation video it was made sure that the participants get the same information about the working mechanism of the Reactable and the objects that are listed above. Another important point in the experimental design was to ensure that the order in which the two automatic tonaliser scenarios were presented would not have any effect on the outcome of the users' opinions. Therefore these two scenarios were presented in different orders interchangeably.

After the tests, the participants were asked to fill out a *post-test questionnaire* (see Appendix B), which asked them to compare the level of understanding, playability, performativity and interest for future experimentation in the three different scenarios. These were also measured on a 5-point Likert scale.

5.2 Results

There were 17 participants who volunteered to do the experiment, which was found to be a sufficient number of users for the analysis of the data. The first observation was made on the answers for the post-test questionnaire as a function of the three different scenarios.

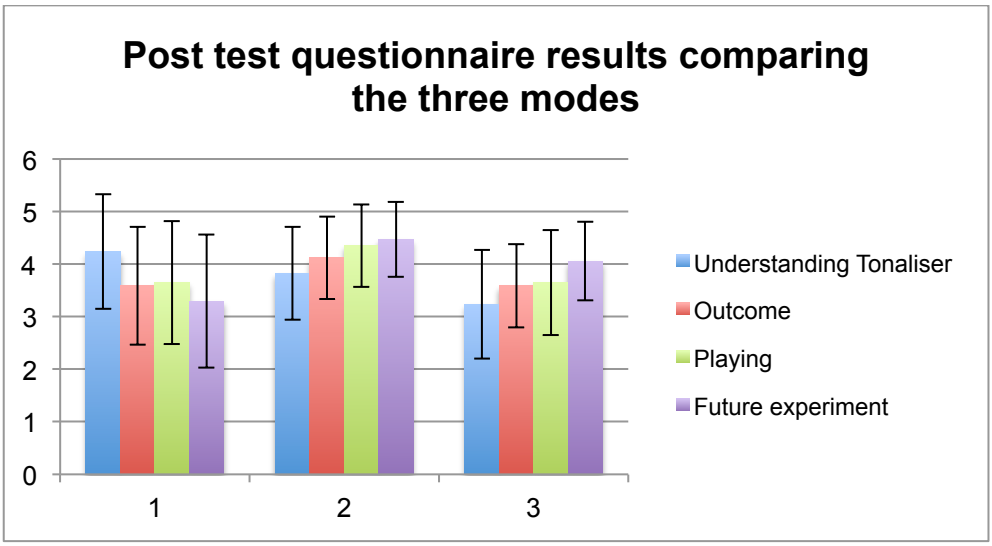

Figure 14. Post-test questionnaire results comparing the three modes with errors as the standard deviation of the data sets

From the above figure it can be seen that the results for evaluating the outcome, the playing and the willingness of further experimentation follow a similar trend. However, the values for understanding the tonality object (blue) in the different scenarios show that this question does not relate to the other ones in the same way. This might be due to the fact that the concept of understanding the tonality object can be thought of in many different ways, whereas the other questions all referred to "liking", a more subjective concept focusing on opinion. Therefore, the further analysis will focus only on the questions that refer to "liking".

The following diagram shows the average of the answers for the questions related to outcome, playing and future experimentation.

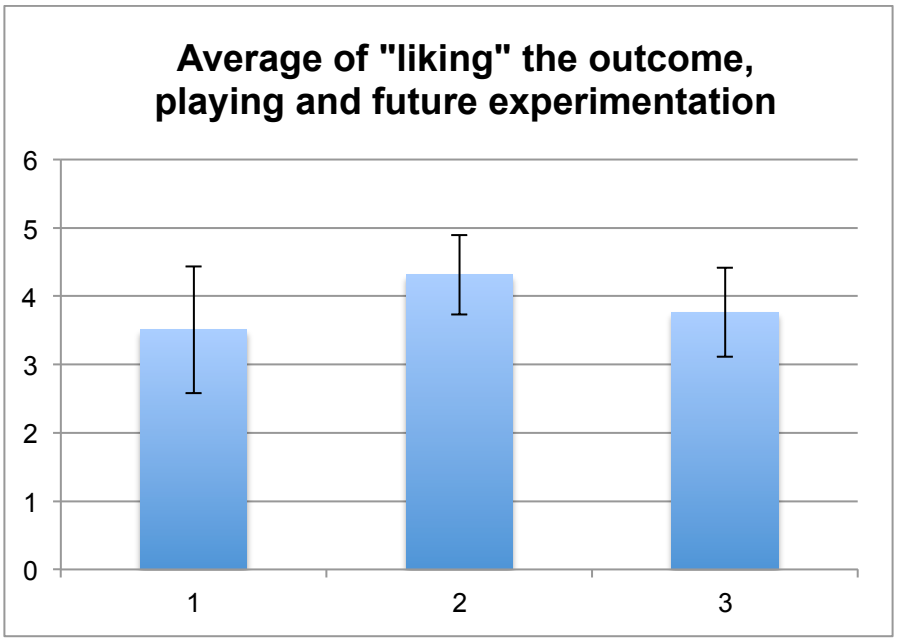

Figure 15. Average of liking the outcome, playing and future experimentation with errors as the standard deviation of the data sets

The above figure shows that the second scenario was the most favoured one with an average of *4.31± 0.58*, followed by the third with *3.76±0.65* and finally the first scenario with *3.51± 0.93* (the error measures refer to the standard deviation of the respective data sets). This result shows that in general, both of the automatic tonaliser scenarios were favoured compared to the manual one. In order to find the significance of these results, a T-Test had to be run on the data sets. From these tests it turned out that the 2nd scenario is significantly larger than both the 1st and the 3rd ($p = 0.0049$ and $p = 0.0145$ respectively) and that there is no significant difference between the 1st and 3rd ($p = 0.362$).

These results show a generalised view on the outcome of the scenarios, however they do not convey any information regarding the observations as a function of musical knowledge. As it was discussed before, this was an important aspect of the measurements.

As mentioned earlier, the pre-test questionnaire assessed the musical knowledge of the participants. The first four questions were designed in a way that their average could be used as a measure for general musical knowledge. This way the average of the preferences of the three modes could be plotted against the musical knowledge for each participant.

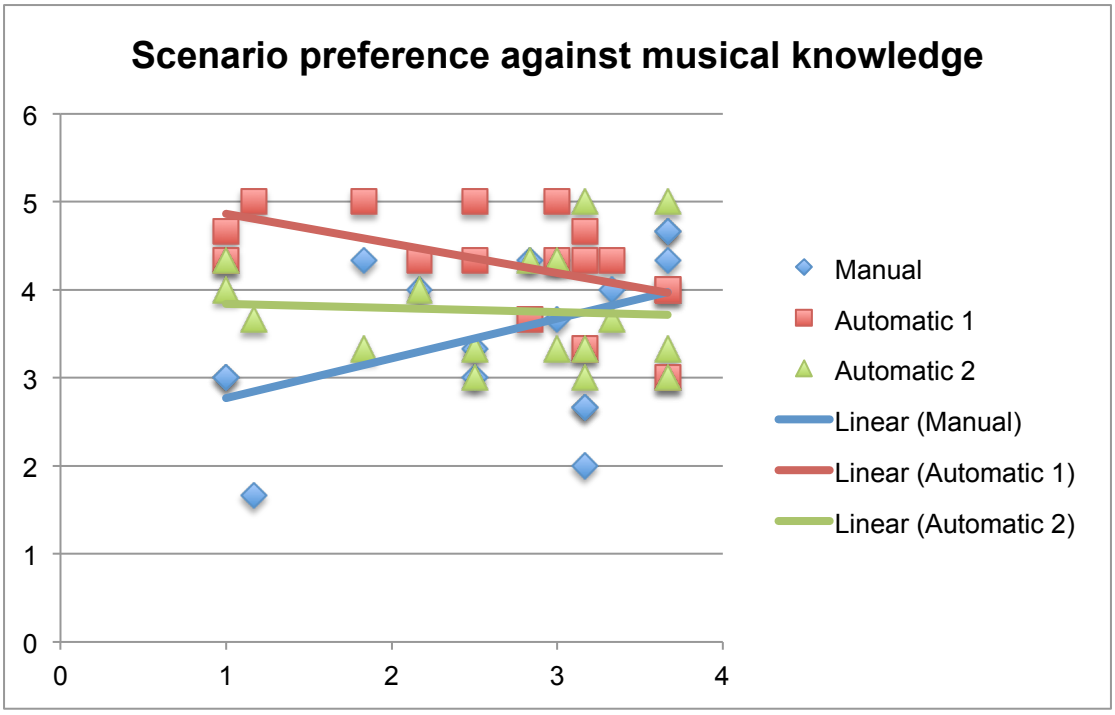

Figure 16. Average absolute preference against musical knowledge

It can be seen that the trend line of the manual version (blue) has a positive correlation - ascending tendency - with musical knowledge, which means that people with higher level of musical knowledge preferred using the manual version. On the other hand, the other trend lines corresponding to the automatic versions (red and green) both have a negative correlation - descending tendency - therefore suggesting the opposite to the previous observation. These results prove that the initial hypothesis was correct, that there is a relationship between musical knowledge and the preference for an automatic musical agent.

Although the manual tonaliser scenario did not serve as a control experiment to which the other two should be compared to, the fact that the subjective opinions could vary between participants suggests that it is worth examining the relative preference between the automatic versions and the manual one. This could be done by subtracting the manual version's results from the automatic ones. This gives a relative difference between the tasks, avoiding biases that could come from the subjective self-evaluation of the participants.

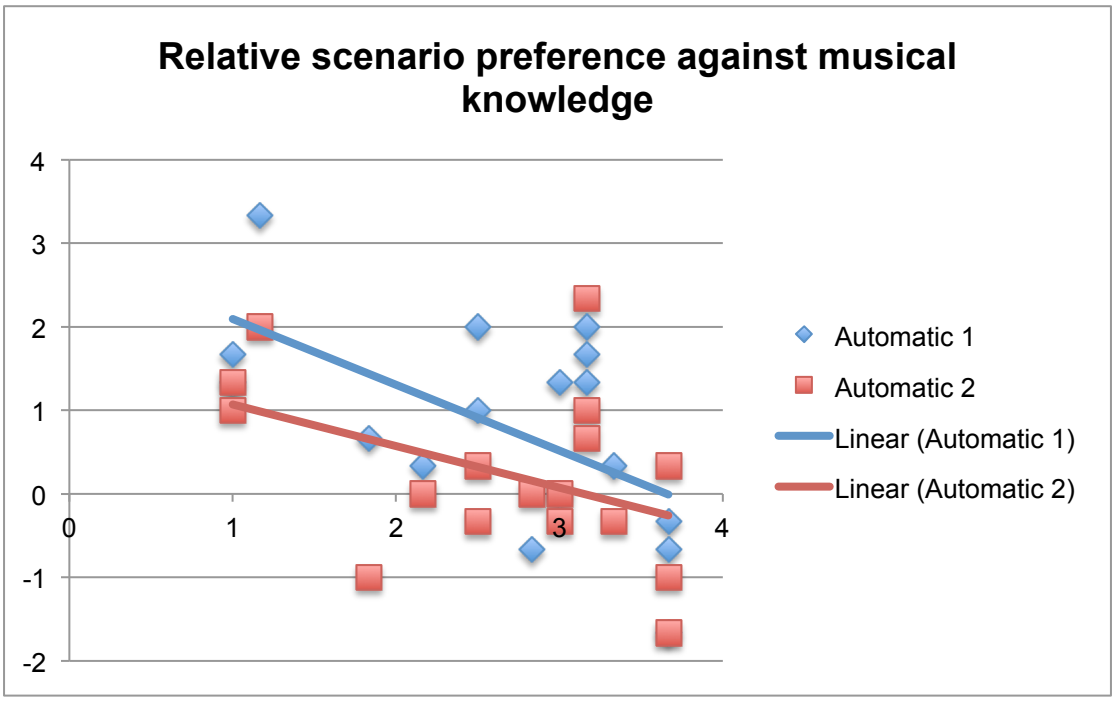

Figure 17. Average relative preference against musical knowledge

It can be seen that the results are similar to the previous diagram for the two automatic versions, however more accentuated. The trend lines have a steeper descending tendency than previously, but the difference between them can still be recognised. This strengthens the observation that users with lower level of musical knowledge preferred using both of the automatic versions, but the implementation for "Automatic 1" is in more favour throughout the entire data set (blue trend line never goes below zero).

6 Discussion

As seen from the analysis of the user tests, it can be concluded that the first version of the automatic tonaliser was preferred the most, followed by the second automatic and finally the manual version. This might be due to a variety of reasons, however some observations can be made that could suggest an explanation for these results.

Firstly, it was very easy and straightforward to see that the users focused on different parts of music performance and production when playing in the different scenarios. When using the manual tonaliser version, they mainly concentrated on selecting the desired notes that could go well with the audio material, focusing on tonality putting other aspects of music making as secondary steps. This approach seemed more adequate for off-line music production, rather than real-time performance. In the case of the automatic tonaliser scenarios, they immediately directed their attention towards the playing/mixing of the audio files and changing between the tonaliser presets. This method appeared to be more suitable for actual real-time performance than the previous one.

From private conversations and subjective observations of the tests it seemed that the first automatic version was favoured more due to the fact that it gave more control to the user. The fact that in this version the different presets were directly related to each other, and the relation between them was obvious to see and hear, gave confidence in using it. These features might be the reason why the users did not like the unpredictable nature of the second version. However - as foreseen before – the musically more trained ones could still see/hear the tonal relevance of the second implementation. From these comments it can be seen that the fundamental reason for preferring the first version to the second relies on a human-computer interaction question and not on the inaccuracy of the information retrieval, neither on an insufficient implementation of the tone generation algorithms. It turned out that the interface, which the users are presented with has a significant effect on their "liking" of playability, outcome and future experimentation.

In relation to these remarks, it is important to see the shortcomings of the implemented algorithms. When creating a musical expert agent, which provides musical knowledge to the user, the aspects of interaction and the way the given musical information is presented is an essential question, which should drive the development of the tool itself.

These observations are fairly preliminary and qualitative, however can serve to pave the way for future lines of development and as corner stones of further experimentations on the subject. In conclusion, the idea of integrating an expert agent into a music production and performance system was found to be successful. Future research could improve the presented implementations as well as spread the areas of development to other fields of musical expert agents, taking into account the above results and discussions.

7 References

[Ableton, 2013] Ableton AG, "New in Live 9/Reclaim the music in your samples…". Retrieved from https://www.ableton.com/en/live/new-in-9/, 2013.

[Assayag et al., 2006] Assayag, G., Bloch, G., Chemillier, M., Cont, A., and Dubnov, S. "OMax brothers: a dynamic topology of agents for improvization learning". Proceedings of the 1st ACM workshop on Audio and music computing multimedia (AMCMM '06). ACM, New York, NY, USA, 125-132.

[Bonada and Serra, 2007] Bonada, J. and Serra, X., "Synthesis of the Singing Voice by Performance Sampling and Spectral Models." IEEE Signal Processing Magazine 24(2):67–78. 2007.

[Chew, 2000] Chew, E. "Towards a Mathematical Model of Tonality". Doctoral dissertation, MIT, 2000.

[Cont, 2008a] Cont, A. "Modeling musical anticipation: from the time of music to the music of time". Doctoral dissertation, UC San Diego, San Diego, CA, USA, 2008.

[Cont, 2008b] Cont, A. "Antescofo: Anticipatory synchronization and control of interactive". Proceedings of the International Computer Music Conference, 2008.

[Cubase, 2012] Bachmann, C., Bischoff, H., Bröer, M., Kaboth, C., Mingers, I., Pfeifer, S., Schütte, B. "Operational Manual, Cubase 7". Steinberg Media Technologies GmbH, 2012.

[Dubnov et al., 2007] Dubnov, S., Assayag, G., and Cont, A. "Audio oracle: A new algorithm for fast learning of audio structures". Proceedings of International Computer Music Conference (ICMC), 224-228, 2007.

[Fiebrink, 2011] Fiebrink, R. "Real-time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance", Doctoral dissertation, Princeton University, January 2011, Princeton, NJ, USA, 2011.

[Fujishima, 1999] Fujishima, T. "Realtime chord recognition of musical sound: a system using common lisp music". In Proceedings of the International Computer Music Conference (ICMC), Beijing, pp. 464-467, 1999.

[Gomez, 2006] Gomez, E. "Tonal description of music audio signals". Doctoral dissertation, Universitat Pompeu Fabra, Barcelona, 2006.

[Krumhansl, 1990] Krumhansl, C. L. "Cognitive foundations of musical pitch". Oxford University Press, New York, 1990.

[Krumhansl, 2000]. Krumhansl, C. L. "Tonality induction: a statistical approach applied cross-culturally". Music Perception, Special Issue in Tonality Induction, 17(4): 461–481, 2000.

[Longuet-Higgins, 1971] Longuet-Higgins, H. C. and Steedman, M. J. "On interpreting bach", Machine Intelligence, 6: 221–241, 1971.

[Maestre et al., 2009] Maestre, E., Ramirez, R., Kersten, S. and Serra, X. "Expressive Concatenative Synthesis by Reusing Samples from Real Performance Recordings", Computer Music Journal 33(4): 23-42, 2009.

[Purwins et al., 2000] Purwins, H., Blankertz, B., and K. Obermayer. "A new method for tracking modulations in tonal music in audio data format". Neural Networks - IJCNN, IEEE Computer Society, 6:270-275, 2000.

[Schnell et al., 2011] Schnell, N., Bevilacqua, F., Rasamimana, N., Blois, J., Guédy, F. and Fléty, E. “Playing the "MO" - Gestural Control and Re-Embodiment of Recorded Sound and Music”, Proceedings of the International Conference on New Interfaces for Musical Expression, pp. 535-536, 2011.

[Schwarz, 2005] Schwarz, D, "Current research in Concatenative Sound Synthesis", Proceedings of the International Computer Music Conference (ICMC), 2005.

[Shepard, 1982] Shepard, R. N. “Structural representations of musical pitch”. In Deutsch, D., editor, The Psychology of Music, First Edition. Swets & Zeitlinger, 1982.

[Temperley, 1999]. Temperley, D. “What’s key for key? the Krumhansl-Schmuckler key finding algorithm reconsidered”. Music Perception, 17(1):65–100, 1999.

[Zils and Pachet, 2001] Zils, A. and Pachet, F. “Musical Mosaicing”, Proceedings of the COST G-6 Conference on Digital Audio Effects (DaFx-01), University of Limerick: 39-44, 2001.

8 Appendix

A Pre-Test Questionnaire

1. I can play a musical instrument
Strongly disagree □ □ □ □ □ Strongly agree

2. I know music theory and harmony
Strongly disagree □ □ □ □ □ Strongly agree

3. I can read/write music
Strongly disagree □ □ □ □ □ Strongly agree

4. I can compose music
Strongly disagree □ □ □ □ □ Strongly agree

5. I play/compose electronic music
Strongly disagree □ □ □ □ □ Strongly agree

6. I know/have played the Reactable
Strongly disagree □ □ □ □ □ Strongly agree

B Post-Test Questionnaire

1. I understood how each object works and how they are related to each other
Strongly disagree □ □ □ □ □ Strongly agree

2.1. I understood how the *Tonality object* works in test 1
Strongly disagree □ □ □ □ □ Strongly agree

2.2. I understood how the *Tonality object* works in test 2
Strongly disagree □ □ □ □ □ Strongly agree

2.3. I understood how the *Tonality object* works in test 3
Strongly disagree □ □ □ □ □ Strongly agree

3.1. I liked the outcome of test 1
Strongly disagree □ □ □ □ □ Strongly agree

3.2. I liked the outcome of test 2
Strongly disagree □ □ □ □ □ Strongly agree

3.3. I liked the outcome of test 3
Strongly disagree □ □ □ □ □ Strongly agree

4.1. I liked playing in test 1

Strongly disagree □ □ □ □ □ Strongly agree

4.2. I liked playing in test 2

Strongly disagree □ □ □ □ □ Strongly agree

4.3. I liked playing in test 3

Strongly disagree □ □ □ □ □ Strongly agree

5.1. I would like to experiment more with the settings of test 1

Strongly disagree □ □ □ □ □ Strongly agree

5.2. I would like to experiment more with the settings of test 2

Strongly disagree □ □ □ □ □ Strongly agree

5.3. I would like to experiment more with the settings of test 3

Strongly disagree □ □ □ □ □ Strongly agree

C User Test Results

Pre Test Answers

Participant	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Average music knowledge (Q1-Q4)
Number 1	5	4	3	3	3	1	3.1
Number 3	5	4	3	3	3	4	3.6
Number 5	3	2	1	5	4	4	3.1
Number 7	3	2	1	2	2	5	2.5
Number 9	1	1	1	1	1	1	1
Number 11	5	3	2	2	2	4	3
Number 13	4	4	3	1	4	4	3.3
Number 15	4	3	1	2	4	5	3.1
Number 17	4	2	2	4	1	4	2.8
Number 2	1	1	2	1	1	1	1.1
Number 4	5	5	3	4	3	2	3.6
Number 6	4	5	5	2	1	1	3
Number 8	2	1	2	1	2	3	1.8
Number 10	1	1	1	1	1	1	1
Number 12	3	3	2	1	1	3	2.1
Number 14	4	3	4	2	1	1	2.5
Number 16	5	5	4	4	2	2	3.6
Total Average	3.2	2.7	2.2	2.1	2	2.5	2.5
StDev	1.4	1.4	1.2	1.3	1.1	1.5	0.9

Post Test Answers

Participant	Q 1	Q 2.1	Q 2.2	Q 2.3	Q 3.1	Q 3.2	Q 3.3	Q 4.1	Q 4.2	Q 4.3	Q 5.1	Q 5.2	Q 5.3
Number 1	3	3	2	2	2	3	3	2	4	3	2	3	3
Number 2	5	5	3	3	5	3	2	5	4	4	3	5	4
Number 3	5	5	4	4	3	4	5	3	4	5	2	5	5
Number 4	5	5	5	4	3	5	4	3	5	3	3	5	3
Number 5	4	2	3	3	3	4	4	3	5	5	3	4	4
Number 6	2	5	4	2	5	5	3	4	5	2	2	5	5
Number 7	5	5	4	3	4	5	3	5	4	3	3	4	5
Number 8	5	5	5	2	4	5	3	2	5	3	2	4	4
Number 9	5	5	4	4	5	4	4	3	4	5	5	3	4
Number 10	2	4	3	1	2	5	4	2	5	4	1	5	3
Number 11	5	5	4	4	5	3	5	5	5	5	4	4	5
Number 12	4	5	3	3	4	4	4	5	4	4	4	5	5
Number 13	4	4	5	4	3	5	3	5	5	3	5	5	4
Number 14	5	5	5	5	3	4	4	3	5	4	3	5	4
Number 15	3	2	3	3	3	4	4	4	4	4	5	5	4
Number 16	4	3	4	4	2	4	3	3	4	3	5	5	3
Number 17	4	4	4	4	5	3	3	5	2	2	4	4	4
Total Average	4.1	4.2	3.8	3.2	3.6	4.1	3.6	3.6	4.4	3.6	3.3	4.5	4.1
StDev	1.1	1.1	0.9	1.0	1.1	0.8	0.8	1.2	0.8	1.0	1.3	0.7	0.7