Interactive Digital Music: Enhancing Listener Engagement with Commercial Music

Toulson, R., Paterson, J., Lever, S., Webster, T., Massey, S. and Ritter, J.

This is a pre-publication version of a book chapter published in Hepworth-Sawyer, R., Hodgson, J., Toulson, R. and Paterson, J.L (eds.) Innovation In Music II, Future Technology Press, UK, pp. 193-209.

Interactive Digital Music: Enhancing Listener Engagement with Commercial Music

Justin Paterson[1], Rob Toulson[2], Sebastian Lexer[1], Tim Webster[2], Steve Massey[1], Jonas Ritter[1]

[1] London College of Music, University of West London, UK.
justin.paterson@uwl.ac.uk

[2] CoDE Research Institute, Anglia Ruskin University, Cambridge, UK.
rob.toulson@anglia.ac.uk

Abstract

Listeners have long been inspired to interact with music and create new representations of popular releases. Vinyl offered many opportunities to reappropriate chart music, from scratching and tempo manipulation to mixing multiple songs together. More recently, artists could engage their audience to interact with their music by offering mix-stems online for experimentation and sharing. With the extended processing power of mobile devices, the opportunities for interactive music are dramatically increasing.

This paper presents research that demonstrates a novel approach to interactive digital music. The research looks at the emergent format of the album app and extends existing paradigms of interactive music playback. The novel album app designed in this research presents a new opportunity for listeners to engage with recorded content by allowing them to explore alternative takes, renditions of a given song in multiple genres, and by allowing direct interaction with embedded mix-stems. The resultant audio remains true to the artist and producer's studio vision; it is user-influenced, but machine-controlled. The research is conducted in collaboration with artist Daisy and The Dark and was funded by the UK Arts and Humanities Research Council.

1. Introduction

This paper presents the work in progress of a research project that was designed to develop a unique interactive music album format that represents a new paradigm in the way we listen to and engage with digital music. The research focuses on identifying and evaluating new modes of engagement between popular artists and audiences, whilst simultaneously identifying a bold new business model for the music industry. In general, digital platforms have been designed in an attempt to replicate and replace the linear analogue music formats; however, modern digital playback systems from mobile devices to networked hi-fi systems are extremely over-engineered for simply 'playing' music and audio. Such devices offer far more creative processing capabilities and are, as yet, underexploited commercially.

A principal objective was to investigate the opportunities for popular music playback to be interactive and malleable, whilst retaining and extrapolating the artistic vision of the musicians and producer. The greater investigation was centred in two specific areas – what are the capabilities of available technologies and the subsequent opportunities for artists, and what is the audience and consumer response to these novel creative approaches? This paper is centred upon the former.

1.1. Modes of Variable Playback

Interactive music can be defined as music in which it is the user's intention to change the nature of the audio, whereas variation in *dynamic music* is primarily a function of some other variable or state [1], for instance user-engagement with a video game. Interactive music can be considered from two different viewpoints. An example of *produced interactivity* enables a listener to manipulate the musical elements of a song in a number of different ways. The song might be an upbeat rock track including drums, electric guitars, vocals and a string section. Consider however, that at any point whilst listening, the listener might prefer to hear the song in a more relaxed manner without disrupting the flow of the music. They may prefer to listen to a stripped down version, perhaps with the electric guitars being replaced by acoustic versions, the drums becoming hand percussion and the strings taking a more prominent role in the mix. Similarly, the listener might select an electronic version of the song, which introduces programmed drumbeats and synthesized sounds. The musical content might have changed, but it is still the same song, and by employing combinations of bespoke alternative audio content created in the studio, the musician's artistic vision is maintained.

There are further opportunities to interact with the music within either a specific and fixed genre, e.g. there could be multiple vocal takes which the listener can choose from, or different guitar or drum parts that represent the song either subtly or dramatically differently. The term produced interactivity is used because the artist still has control of the boundaries for playback, deciding what genres and instrumentation they create as a palette, and they will also define the default playback setting – the version released for conventional sale and distribution.

Algorithmic music is well established and in this, an algorithm maintains full control over what variation of the audio content is being played back; this might include motifs, timbres, rhythms and other musical content. *Algorithmic interactivity* can represent a form of interactive music in that it is processed predominantly with input from mathematical and statistical algorithms, but with user-influence on the result. The listener might select genres and playback styles as with the produced method, however this system responds subtly differently. For example, a number of gestures that a user might make could be automated into a unique sequence, or algorithmic components might include preset modes of real-time processing, enabling the listener to trigger the application of these at will.

1.2. Further Aspects of the Project

It is also necessary to consider the tools required for musicians, music producers and importantly, record labels to develop their own interactive music. This requires a content management system (CMS) that allows consistently repeatable population of the format – an operational paradigm with a simple user interface design that facilitates building of bespoke interactive music releases. Standard industry procedure involves matching the relative loudness of tunes in an album at the final mastering stage. A large number of different combinations of audio files in combinations that might change dynamically are a heterodox to this. Additionally, in bringing interactive music systems to a diverse audience with different preferences and behavior patterns, it is useful to be able to gather data on the nature of the user response to this new medium. The last two areas are beyond the scope of this paper.

2. Background and Related Work

2.1. Game Audio

Computer games have featured dynamic (often referred to as *adaptive*) music since the 1980s. LucasArts used their iMuse engine [2] in many of their games. This system chose from multiple MIDI sequences that were triggered on demand by a number of pre-programmed decision points that could steer the compositional direction through a ‘tree’ of sequences. This developed into horizontal re-sequencing systems that re-ordered pre-composed blocks of music in response to the user’s gaming decisions and situation [3]. Vertical re-orchestration utilized a number of concurrent layers of music which could be added to or taken away, again in response to gaming activity [3].

Albeit in the context of games, in 2012 Berndt et al. highlighted the negatives of static and repetitive music, and produced a survey of approaches to remedy this with dynamic technologies, pointing out that “neither fully static compositions nor completely unrelated (random) music do an expedient job” [4, p. 62]. This paper implies the readiness of the modern consumer for greater variation in the listening experience and further, looked forward towards the development of more sophisticated audio engines that might deliver this. In 2014, Gasselseder [5] produced a scientific study which confirmed that game players experienced a greater degree of immersion when experiencing dynamic music.

2.2. The Album App Format

One emerging format of music delivery is the mobile application (often referred to simply as an *app*). The *album app* format is valuable since it allows unique artistic and interactive content to be distributed alongside a collation of audio, supporting the notion that an album is more than just a collection of songs, but potentially a representation of artistic vision which may include artwork, photography, lyrics, video, animation, gaming, social networking and crucially – interaction. The album app is also an incredibly attractive method for music delivery because it is secure, i.e. once the app is created it generally cannot be tampered with, ensuring both digital integrity (as opposed to MP3s) and added piracy resilience, since apps are much harder to duplicate and distribute than simple audio files. Ultimately, the concept of the format could extend into adoption by smart TVs and games consoles. Streaming music services are nowadays also adding additional media content to albums, such as song lyrics and extended artwork. It is therefore possible that album apps and audio streaming applications could become merged in future innovations.

2.3. Previous Interactive and Album App Releases

Bands such as Rush have previously offered combined (static) music and media apps as far back as 2010, and in 2014 Paul McCartney re-released five of his solo albums as album apps; however, few interactive music applications have been released in recent years. Bjork's 2011 *Biophilia* album is perhaps the most recognized to be released to date. Alongside music, artwork, credits and animations, the app also introduces interactive elements for each song, allowing the user to engage with different aspects of sound and visualization with respect to the album content. Similarly Peter Gabriel's *Music Tiles* app, although not an album itself, allows the user to interact with the music and produce unique (albeit basic) mixes of the tracks, and is designed to be more of a game than a playback medium. Jorge Drexler's *n* app also features songs written specifically for interactive playback, attempting to utilize the full processing capabilities of digital mobile devices. Gwilym Gold's *Tender Metal* used an algorithmic system to play back synthesized backing differently every time, although without user control. The Jammit series [12] of teaching tools that started in 2012 allowed muting and volume control of the individual stems, slowing down of the music, notated transcriptions and recording of the user playing along. Pitbull's *Planet Pit* offers many interactive multimedia features, but ultimately is a promotional tool since it only features excerpts of the actual music. 2013 saw the release of DJ Vadim's *Don't Be Scared* and *This DJ* via the Immersive Album LTD system. These are apps that play back video and allow users to engage with features on the screen and change elements of the music and video simultaneously, effectively substituting audio stems to change vocalist or drumbeat. The company Reactify [16] have a number of apps including *VW & Underworld: Play The Road*, in which the interaction comes not from the human, but from whatever is being done with the car to control a live remix of the band Underworld.

Many new developments happened in 2014. Reactify also released the remixing app *CTRL*, which allowed a number of time-synchronized DJ effects to be applied to the tracks of a number of popular artists. Shakhovskoy and Toulson [17] defined a potential album-app model with artist Francois and the Atlas Mountains for their album *Piano Ombre,* developed by Script. It was regarded as the world's first chart-eligible app and is recognized as the foundation software platform for which the interactive app described in this paper is built upon. Bernhoft Islander released his self-titled album as an app, including interactive features that allow the listener to manipulate the balance and panning of instrument stems and to experiment with looping motifs and phrases from his songs [18].

2.4. Multi-Track Music Formats

A number of multi-track audio formats have been created, and have been documented by Redhead [19] and Taglialatela [20] amongst others. iKlax [21] is a file format that allows the clustering of up to 10 audio tracks (in a mobile app – more on a desktop machine) and offers the developer the chance to mute, solo and adjust the volume of the individual tracks. iKlax has been implemented in apps such as *Perform A Track*, which is targeted at those wishing to play along in a *music minus one* [22] fashion. The Audizen company developed a similarly specified system to iKlax called *Music 2.0*, and in 2010, the best features from each were adopted into the MPEG-A Interactive Music Application Format (IM AF). This was specified [23] to offer: multiple audio tracks and a definition of their hierarchical structure, data that pre-defined mixing information and rules that introduce user-interaction data, along with timed text, images and metadata. Any interactive music player that supported this format would be able to offer both preset 'producer' versions of songs, and also allow users to create their own unique mixes.

Various other formats have also been created. MXP4 was released in 2009, and this offered video streams and consumer-focused metadata such as artist biographies. Another multi-track format is the open-source MOGG [24], which supports multiple Ogg Vorbis files, although it was not widely adopted. Song Galaxy also produced a multi-track format called MTF (Multi Track File) [25] targeted at the karaoke market, but also allowing certain remix-type actions, including transposition. Yet another format is iXMF, which is a wrapper for both audio and MIDI files, and contains cue sheets that allow specific events to be triggered at a particular point in time. The MIDI/audio engine that powers *Tender Metal* [26] is the BRONZE Format [27], developed at Goldsmiths University in 2012. In 2015, Native Instruments released *Stems*, a DJ tool that offers manipulation of four separate tracks, packaged in a MPEG-4 Container [28]. This system works with their Traktor Software, and also comes with a CMS that allows musicians to create and package their own material for subsequent manipulation.

3. Research Methodology and Application Design

3.1. Research Objectives

The key project objectives and research questions were defined by the overriding aim to investigate new ways that contemporary music-playback platforms can potentially:

1. Support novel playback paradigms that can engage listeners and hence generate new revenue streams for the commercial music industry

2. Create a more intimate and creative connection between the artist and their fan base

In order to evaluate these research objectives, a methodology to design, build, test and evaluate a new interactive digital music platform was identified, as shown in Figure 1.

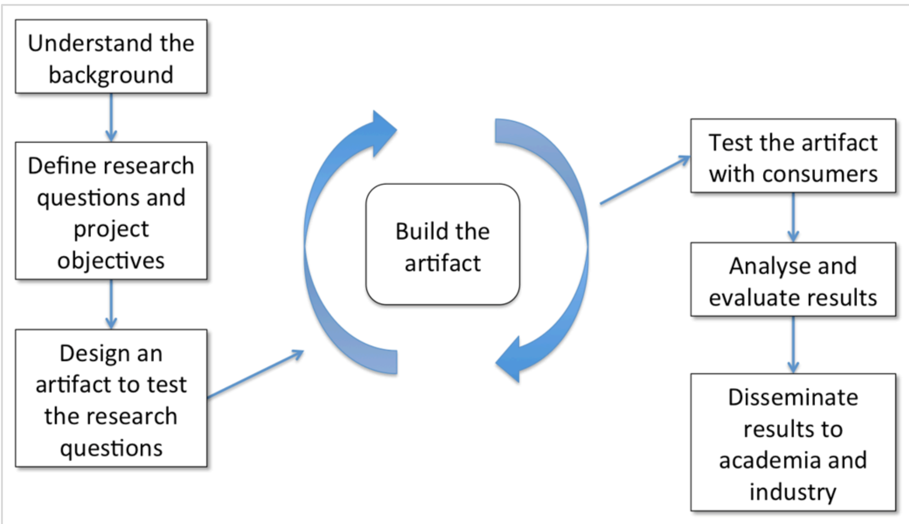


Figure 1. Research methodology for the album app design project

3.2. Application Design and Implementation

The development team agreed functional design specifications during an early brainstorming stage of the project, and these were then implemented in the agile and iterative development build-cycle shown in the centre of Figure 1. The iterative development process is shown in more detail in Figure 2, which also represents the seven-month app-development cycle. The initial brainstorming and prototyping phase was intended to evaluate a number of interactive playback paradigms that could potentially be used in a mobile music application. The prototyping stage was conducted in a number of high-level audio manipulation packages including Max/MSP, Pure Data, Logic and Abelton Live, as well as engineering packages including Matlab and Xcode. On development of the high-level prototypes, a review process, including feedback from the collaborating artist Daisy and The Dark, was required to decide which specific prototypes would be taken forward to the final application design and developed in the Objective-C language (in Xcode) for implementation on Apple iOS devices, specifically the iPhone and iPad.

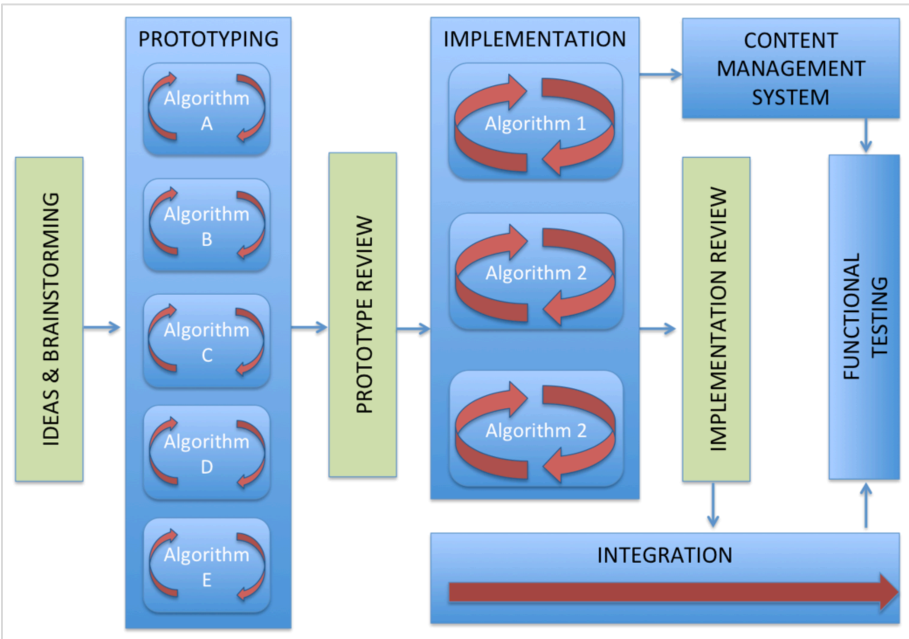


Figure 2. Iterative application development process

The integration phase of the project involved the repurposing of the album app platform originally developed by project collaborator Script [17]. This was to implement the new interactive music features and to populate the platform with the audio and visual assets of Daisy and The Dark. Alongside the integration phase runs a prototype CMS development process, which is intended to define and test the features that are required in order to develop a platform that any music artist or record label might populate with their own audio and visual media.

The functional design for the interactive album app's music playback features is built upon a three-layer architecture. The architecture defines the linking of *graphical user interfaces* (GUIs) with an advanced *audio playback engine*, via a *rulesets* layer that defines the specific interaction operations and the user controlled features, as shown in Figure 3.

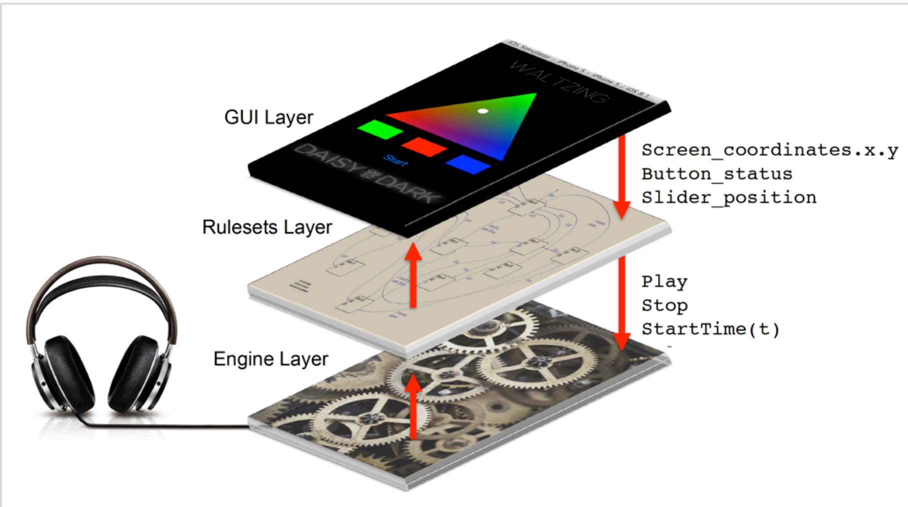


Figure 3. Interactive digital music application architecture

Figure 3 shows that user-interaction messages, such as those representing touch locations of screen co-ordinates, button presses and control slider positions are then evaluated by the specific rulesets and used to request specific audio playback and digital signal processing actions from the audio engine. Additionally, the audio engine reports its status to the rulesets layer, which informs the GUI layer to update whenever audio playback is modified, such as a new track starting or when an algorithmic interaction is implemented.

4. Functionality and Operation

4.1. Audio Playback Engine

The design concept of the audio engine is to allow simultaneous playback of 36 audio files, which is the maximum number imposed by the CPU capability of the iPhone 5, the earliest device for which compatibility was sought. A given song uses six stem families, for instance: vocals, guitar, bass, drums, strings and keys, although the actual instrumentation is different for each song and song version. This allows 36 combinations of the six stems to be available to the user in real time, although only six could be heard at any one point. This was achieved by the implementation of a 36-track audio player, designed using the Audio Unit *AUFilePlayer* library – part of Aple's iOS.

Developed in the Objective-C programming language in the Apple Xcode programming environment, the audio engine consists of two major bespoke components (classes): the *SLAudioEngine* object and the *SLtimer*. The SLAudioEngine implements 36 AUFilePlayers, taking their source audio as specified by a file URL supplied in the *songName.plist*. These players connect to a 36-input *AUMixer* object, which facilitates control over audio playback. The SLtimer controls synchronization and scheduling.

Within in the SLAudioEngine are fade engines that allow the design of *transitions*, i.e. the timed fade to a defined volume level, with one for each stem. Transition profiles (are stored in the songName.plist preset file and are created by using the custom-built CMS system – the user selects suitable profiles from a pre-defined palette. Each transition is defined by a mix level in dB, and the signal is assumed to be silent if it is at a level below –99 dBFS and is flagged accordingly. Each transition allows either a linear or equal-power fade curve, and steps through 100 increments within a definable time. Fades can start on any of these steps, as well as stop at any step greater than the start.

The current implementation of the algorithm allows CMS-user-definition of 36 transitions. Upon triggering by the end-user's gesture, each fade engine evaluates whether the current volume of the corresponding mixer input needs to be adjusted to reflect the volume level defined in the specified transition. If the defined volume level is greater than the current volume level, an increase will occur over the range between the defined start and end steps. Conversely, if the desired volume value is lower, a fade out will occur between the defined steps.

As the transition time is variable, the app GUI may supply and influence the transition time. The CMS also allows quantization or limiting of the moment of triggering within a calibrated range of settings, to force transitions to occur in time with the tempo of the song and in a musically appropriate fashion. Transitions can also be triggered by intelligent computer control, either randomly or in response to some other algorithmic event.

4.2. Interactive User Interfaces

A number of interactive GUIs were prototyped and four were chosen to be included in the final application. These were those referred to as *stem faders, stem switches, manual mix crossfades* and *intelligent mix crossfades*.

The most simple user interfaces were those that allowed fading and on/off switching of audio stems. Figure 4a and Figure 4b show the *stem faders GUI* for a single song (in this case the song *Red Planet*). Each vertical slider allows manipulation of the playback volume of a particular audio stem. For example, *Red Planet* is made up of audio stems representing drums, bass, synthesizers, cellos, strings, and vocals.

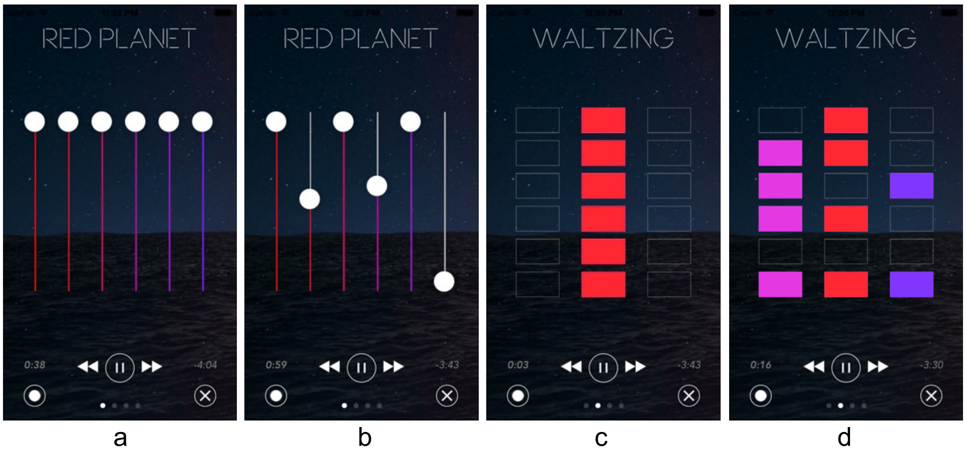


Figure 4. Stem faders and stem switches GUIs

It is apparent that the user could create a unique mix of the stems to their own preference, but it is also possible for a single instrument stem to be muted, in order for the listener to sing/play along. It is also possible to mute multiple stems and leave a single stem active in solo, e.g. the listener might wish to solo the bass guitar in order to learn how to first mimic, then ultimately play along with the full track, replacing the original part.

The *stem switches* implemented on the GUI for the song *Waltzing* allows the user to manipulate three different versions of the same song. When the six central stem buttons are active (as in Figure 4c), the main radio mix of *Waltzing* is heard (stems representing two piano lines, cello, synthesizer, electric guitar and vocals). An electronic remix is controlled by the left column buttons (drums, bass, piano, synthesizer, electric guitar and vocals) and the right columns activate an orchestral version (made up of different cello, violin and vocal lines). The user can chose to play back one of the three produced versions of the song, or they can manipulate their own mix made up of a combination of the stems across all three song versions (as shown in Figure 4c and Figure 4d). The stem fader positions also influence all different combinations of the stem switches.

The *manual mix crossfades GUI* design is implemented for both the *Ghost* and *Red Planet* songs, though in different geometries, as shown in Figure 5a and Figure 5b. *Ghost* uses a triangular cross fade mixer which incorporates three versions of the song. When the small white cursor (the *mix control*) is at the top point of the triangle, the main radio mix of the song is played back, whereas the bottom corners engage acoustic and electronic versions of the song. The listener can choose to move the mix control towards any corner to a position that blends between two or all three versions. Tapping the mix control toggles the vocals on and off, effectively allowing an instrumental version of the song to be played, although again the stem faders can also influence operation.

The *Red Planet* GUI (Figure 5b) works similarly, though with a circular design. The main radio mix of *Red Planet* is heard when the mix control is positioned in the center of the circular interface. Acoustic, electronic, dub and choral interpretations are positioned at 0, 90, 180 ad 270 degree positions on the circle perimeter.

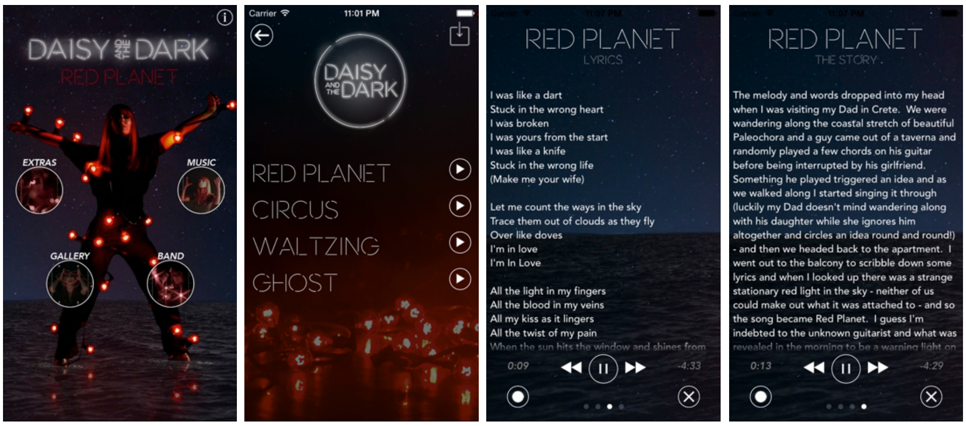
a b c d

Figure 5. Manual and intelligent crossfade GUIs

The Circus user interface is an *intelligent mix crossfade GUI*, implemented as a matrix of 25 different cells, each representing a different mix (as shown in Figure 5c). Different mixes can be engaged by tapping or dragging around the matrix interface. Initially the mix control is positioned in the center, which is the position for the main radio mix of the track. Each corner of the matrix represents an alternative mix of Circus, including acoustic, and electronic remixes; towards the corners, each matrix cell plays a subtly different mix. When a new mix is selected, a unique cross-fade algorithm smoothly transitions the music from one mix to another. The Circus interface also has a unique *variPlay* feature, which is activated by any of the four buttons beneath the main matrix (as highlighted by Figure 5d). When variPlay is active, a quadrant of the matrix is highlighted, and from it, an intelligent algorithm automatically selects new mixes at various points in the song, ensuring that a unique experience is heard on each listen. There are four variPlay modes, each presenting a different quadrant of the matrix.

4.3. Additional Album App Features

The app has a number of additional rich media features that are intended to enhance the listener's overall user-experience. The main home screen (Figure 6a) and home menu allows the user to select different features of the app by dragging menu items to the center of the screen. This launches a short animation and opens the selected app page. The music album page (Figure 6b) gives access to the four tracks of the EP. Any song can be selected, which automatically launches the associated song-player user-interface.



a b c d

Figure 6. App home screen, music album, song lyrics and song description

While in the song-player interface, the user can also swipe left and right to access additional features including the song lyrics (Figure 6c) and a detailed narrative on the song and its origin (Figure 6d).

The band section of the app (accessed from the main home screen) allows the user to read the full artist biography of Daisy and The Dark, as well as scroll through a number of artist images. Additionally the user can access performer and production credits for the EP. A gallery section contains a number of images from the album artwork, and live performances by the artist. A extras section also links to a number of and social network related features.

4.4. Data Collection

Data analytics are built into the app using the propriety Flurry protocols [20]. The data analytics allows collection of user interactivity for a number of aspects of app engagement, menu selections and song interaction. It is therefore possible, with enough app users, to gather meaningful quantitative data on the sections, which receive the most user engagement and the duration spent by the user on each app feature. Additionally it is possible to gather and analyze data of which songs were played the most and the frequency of interaction with individual GUIs and playback features.

The app also includes a built-in research questionnaire which assesses the listener's user-experience in both a qualitative and quantitative manner. Questions include asking users to rate the interactive album app features, to comment on their overall impression of the app, and whether they would be willing to purchase commercial music in this format in the future.

4.5. Content Management System

A CMS is required to enable content providers such as music producers or record labels to integrate their audio into a final compiled app. The CMS allows its user to: specify combinations of stems to be played, trim volumes, set parameters for the fade times between audio stems and to define how the app reacts when different buttons are pressed. The transitions between different song sections could also be auditioned, and variPlay structured. When doing all this, it is essential that the user is able to get instant results that emulate the behaviour of the final app, both to provide feedback to fine-tune the mix and enable experimentation.

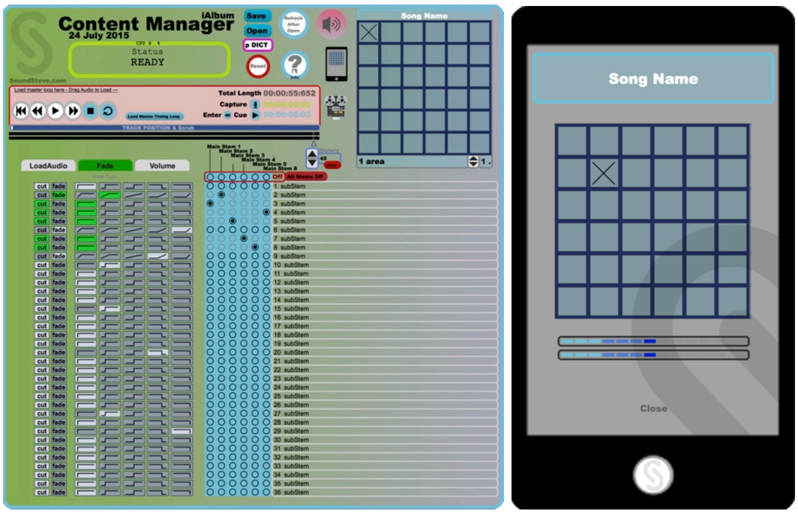


Figure 7. The CMS GUI and iPhone simulator window

The *Content Manager* was created in Max/MSP because this rapid-prototyping program is versatile enough to control audio, capture multi-touch data and format it into a JSON file that could be read by the app. The GUI can be seen in Figure Figure 7 (left), and one of a number of pop-up sub-windows that represents an iPhone user's view (right).

5. Discussion and Conclusions

The app holds data for a total of 96 audio stems and it was chosen that 256 kbps MP3 data compressed files - as opposed to lossless pulse-code-modulation (PCM) - data, would be used to keep the total size of the app to a minimum. The total app size when downloaded from the iPhone app store is 790 Mb, which is slightly smaller than Bjork's *Biophillia* app that has a data size of 849 Mb.

Most commercial iOS apps play back audio using AVPlayers from the AVFoundation library. The AVPlayers only offer non-synchronous playback, and so were not suitable for the 36-track parallel architecture required here. It was for this reason that a low-level (Audio Unit) audio engine had to be designed with bespoke players. The SLAudioEngine also proved much more CPU efficient. The multi-track audio formats discussed in Section 2.4 where evaluated and deemed unsuitable for a context such as this due to lack of controllability in the highly parallel architecture.

There is an opportunity to perform future listening tests to quantitatively evaluate the sound quality loss of using data compressed audio files in this application.

However, audio testing by the development team revealed that the overall quality of mixing stem files in MP3 format to a stereo monitoring stream proved noticeably superior to mixing down the equivalent PCM audio stems and converting the result to MP3. This result is unsurprising given the nature of data reduction in the MP3 codec, which appears to work much more transparently on (unmixed) instrument files. Similarly, the loudness levels of all audio files in the app have been carefully controlled in order to give a smooth sonic experience for the user. There are a number of interesting observations from a loudness analysis perspective, which will therefore be quantitatively evaluated and disseminated in future research publications by the investigating team.

A number of different interactive music GUIs have been developed for the app. The Stem Fader mode of operation is in line with that of some of the apps referred to in Section 2.3, yet maintains relevance here as part of a varied suite. The GUIs all behave rather differently, and are likely to appeal to different people, and again, this aspect will be evaluated in future research tasks. In particular, a full and thorough analysis of the Flurry-gathered user data will be conducted in order to reveal which GUIs have been engaged with the most. This exercise will be supported by the running of focus-group workshops which will allow users to both feed back directly and by completing the built-in app questionnaire. The full evaluation of the user-experience will therefore be disseminated in future publications by the investigating team.

As demonstrated in Section 2.3, to date there are few album app releases and defining a unifying format for the future could hold potential benefit before too many variants emerge. Of course, interactive music is in its naissance and its evolution is likely to involve in a number of different paradigms of varying success, however, this research does offer a functional model that could be developed to evolve further. Commercial exploitation of any such single format would also increase profit margin as releases proliferate, since development costs will have already been covered. Further, lead times to release will be dramatically reduced.

In particular, there is an opportunity to develop more advanced graphic designs that animate alongside the sonic experience, utilizing visual programming techniques that have been developed within the mobile games industry. Interestingly, the games development community has experienced difficulties currently associated with cross-fading between disparate pieces of music, as Gungormusler notes: “one major drawback of the approach is the lack of smooth transitions between the different selections of musical pieces, which is said to be left as future work.” [21, p. 3] It is therefore conceivable that the audio mixing architecture and associated algorithms developed in this research might also serve valuable purpose in future games audio systems.

6. References

[1] K. Collins, Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design. MIT Press, 2008.

[2] M. Z. Land and P. N. Mcconnell, “Method and apparatus for dynamically composing music and sound effects using a computer entertainment system,” US5315057 (A), 24-May-1994.

[3] K. B. McAlpine, M. Bett, and J. Scanlan, “Approaches to Creating Real-Time Adaptive Music in Interactive Entertainment: A Musical Perspective,” presented at the Audio Engineering Society Conference: 35th International Conference: Audio for Games, 2009.

[4] A. Berndt, R. Dachselt, and R. Groh, “A Survey of Variation Techniques for Repetitive Games Music,” in Proceedings of the 7th Audio Mostly Conference: A Conference on Interaction with Sound, New York, NY, USA, 2012, pp. 61–67.

[5] H.-P. Gasselseder, “Dynamic Music and Immersion in the Action-adventure an Empirical Investigation,” in Proceedings of the 9th Audio Mostly: A Conference on Interaction With Sound, New York, NY, USA, 2014, pp. 28:1–28:8.

[6] Jammit, “Jammit | Jam with your Favorite Band.” [Online]. Available: http://www.jammit.com/. [Accessed: 06-Aug-2015].

[7] “Reactify,” Reactify. [Online]. Available: http://reactifymusic.com/portfolio/. [Accessed: 05-Aug-2015].

[8] J. Shakhovskoy and R. Toulson, “Future Music Formats: evaluating the ‘album app,’” Journal on the Art of Record Production, no. 10, Jul. 2015.

[9] D. Greeves, “Can The App Save The Album?,” Sound On Sound, Oct-2014. [Online]. Available: http://www.soundonsound.com/sos/oct14/articles/apps.htm. [Accessed: 06-Aug-2015].

[10] T. Redhead, “Composing and Recording for Fluid Digital Music Forms,” JARP, no. 10.

[11] C. Taglialatela, “MPEG-A Interactive Music Application Format (IM AF) Encoder,” Seconda Universita’ Degli Studi di Napoli, Naples, Italy, 2012.

[12] “iKlax multitrack music file format - Get the SDK for encoding and decoding.” [Online]. Available: http://www.iklax.com/. [Accessed: 05-Aug-2015].

[13] Music Minus One, “Music Minus One - musicminusone.com.” [Online]. Available: https://www.musicminusone.com/. [Accessed: 06-Aug-2015].

[14] MPEG, “Interactive Music Application Format.” [Online]. Available: http://mpeg.chiariglione.org/standards/mpeg-a/interactive-music-application-format. [Accessed: 05-Aug-2015].

[15] “MOGG Files - Multitrack Digital Audio Format,” MOGG Files - Multitrack Digital Audio Format. [Online]. Available: https://moggfiles.wordpress.com/. [Accessed: 05-Aug-2015].

[16] “Music Software Song Galaxy Online Shop,” Song Galaxy. [Online]. Available: http://songgalaxy.com//software.php. [Accessed: 05-Aug-2015].

[17] “Gwilym Gold.” [Online]. Available: http://gwilymgold.com/. [Accessed: 12-Feb-2014].

[18] M. Grierson, “BronzeFormat,” 2012. [Online]. Available: http://bronzeformat.com/. [Accessed: 12-Feb-2014].
[19] Stems-Music, “Stems is for DJs,” Stems, 2015. [Online]. Available: http://www.stems-music.com/stems-is-for-djs/. [Accessed: 05-Aug-2015].
[20] magicspark, “Flurry Analytics,” Flurry, 28-Mar-2014. [Online]. Available: http://www.flurry.com/solutions/analytics. [Accessed: 10-Aug-2015].
[21] A. Gungormusler, N. Paterson-Paulberg, and M. Haahr, “barelyMusician: An Adaptive Music Engine for Video Games,” presented at the Audio Engineering Society Conference: 56th International Conference: Audio for Games, 2015.