

# **BitBox!: A case study interface for teaching real-time adaptive music composition for video games**

McAlpine, Kenneth 2016

This is the accepted manuscript © Ingenta 2016. The article is published in final form in the *Journal of Music, Technology & Education* 9(2). available online: [https://doi.org/10.1386/jmte.9.2.191\\_1](https://doi.org/10.1386/jmte.9.2.191_1)

## **BitBox!: A case study interface for teaching real-time adaptive music composition for video games**

**Keywords:** Adaptive music, video games, musical interfaces

### **1. Abstract**

Real-time adaptive music is now well-established as a popular medium, largely through its use in video game soundtracks. Commercial packages, such as fmod, make freely available the underlying technical methods for use in educational contexts, making adaptive music technologies accessible to students. Writing adaptive music, however, presents a significant learning challenge, not least because it requires a different mode of thought, and tutor and learner may have few mutual points of connection in discovering and understanding the musical drivers, relationships and structures in these works. This article discusses the creation of ‘BitBox!’, a gestural music interface designed to deconstruct and explain the component elements of adaptive composition through interactive play. The interface was displayed at the Dare Protoplay games exposition in Dundee in August 2014. The initial proof-of-concept study proved successful, suggesting possible refinements in design and a broader range of applications.

### **2. Introduction**

This article tells the story of ‘Bitbox!’, an interactive gestural digital musical interface (DMI) designed to deconstruct and explain the underlying technical frameworks that support adaptive video game soundtracks, expose the creative challenges faced by video game soundtrack composers, and provide a framework for exploring the creative methods that underpin them.

BitBox! was created in response to a student need, identified across several years of researching and teaching adaptive music as part of a suite of specialist degree programmes at the University of Abertay (UAD). UAD currently delivers a number of degree programmes focused on the computer games industry. One of these, a Bachelor of Arts in Creative Sound Production, concentrates on the content creation of digital sound and music assets and their implementation. The programme launched in 2006, and, in the years since, it had become clear that while many students have a very keen sense of musical awareness; a good contextual knowledge of the contemporary video game industry; well-developed composition skills, and were keen to study music composition for games, very few of them had any real conceptual understanding of the processes that underpin real-time adaptive music or how to approach the problem of scoring for interactive media.

This, perhaps, should not come as too much of a surprise. Firstly, Game Studies, the field of study that focuses on game design, players, and their role in society and culture, is a relatively new and evolving discipline (Williams, 2005), and the study of the subfield of game music is yet more nascent.

There are a great many similarities between film and game soundtracks, and arguably both share a common purpose. Ever since a pianist accompanied a series of Lumière Brothers' shorts in the Grand Café, Paris on 28th December 1895, music has been integral to the cinematic experience, serving to highlight and drive the narrative, provide insight into characters and heighten the film's emotional impact (Brown, 1995). In the years that followed, conventions that firmly bond screen visuals and music were established and refined, forever linking music with the moving image in the minds of viewers. Little wonder, perhaps, that the first synchronised film soundtracks, *Don Juan* (1926); *The Jazz Singer* (1927), and *Steamboat Willie* (1928), were music-, rather than dialogue-led.

One reason that this marriage has been quite so successful is because both film and music are fixed, fairly linear structures: once a film sequence has been cut, it provides a fixed narrative framework around which can be built the underscore. It provides definite hitpoints to which the composer can work, and is suggestive of tempo and intensity, creative building blocks that translate well into the musical domain, and which can be built upon by a soundtrack composer. Indeed, tried-and-tested methodologies have been well documented over the years. Conceptual approaches can be found in, for example, Adorno & Eisler (1947), while Lalo Schiffrin (2011) has recently published a detailed process and musical analysis of some of his most notable film scores.

Interactive entertainment titles, on the other hand, present fundamentally different challenges that make the tight integration of visual and musical components much more difficult, even though the soundtrack might be intended to function in a similar manner and deliver similar emotional and narrative ends. In a typical first-person-perspective computer game, for example, which provides the player with the illusion of complete freedoms of choice and movement (King & Krzywinska, 2006, p79), it is not possible to predict in advance the particular game-state at any point in time, making the accurate prediction of music cues and hitpoints near-impossible. Similarly, different games feature different game mechanics, the methods the programmers use to facilitate interaction with the game state from which gameplay emerges, and different player types will respond to these mechanics in different ways (Quick, Atkinson & Lin, 2012). Very goal-focused players, for example, will typically focus on completing a set of very specific tasks within the game, often completing each level or task very quickly. Other players might choose to

explore each level more thoroughly, completing tasks in a different sequence and over a much longer timescale as they try to uncover hidden features and objects within the level. The game, of course, has to cater for both types of player, and, indeed, many other types, meaning that the ordering and sequencing of soundtrack hit- and sync-points, and the duration of musical cues, transitions and segues can be as unpredictable as their specific timing.

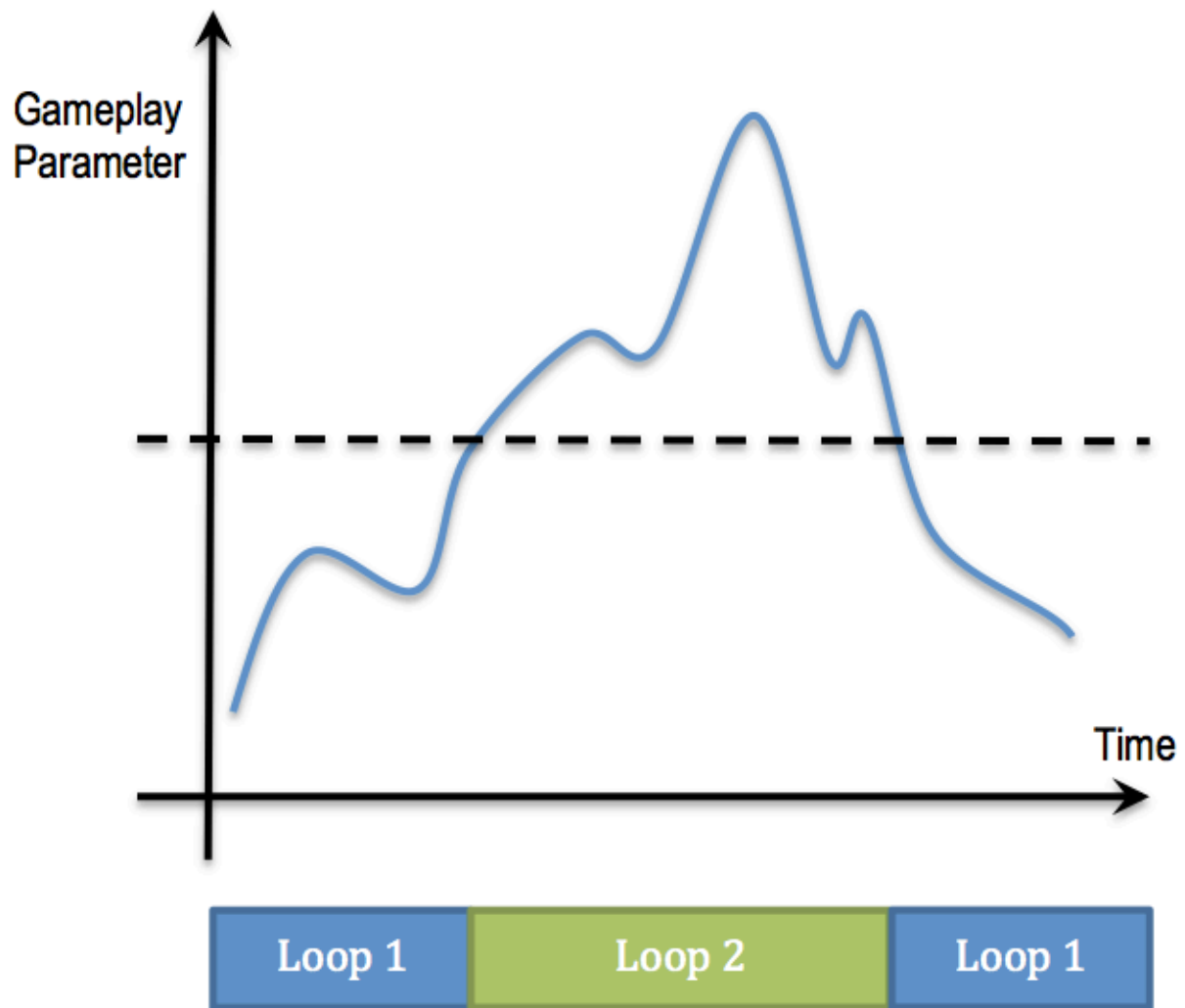
The enormity of these combinatorial possibilities makes it impossible for the composer to anticipate and score for every eventuality, rendering many of the established approaches to scoring for film largely redundant, and a range of adaptive music techniques have grown up to replace them (McAlpine, Bett & Scanlan, 2009).

The simplest of these is the event-driven cue. Here, a game engine – the software framework that provides the core functionality of a computer game – monitors the current game state for a specific game event, such as losing a life. With each event is associated a specific musical cue, such as the six-note melodic sting which is triggered in Super Mario Bros (1985) when the titular hero collects one hundred gold coins and is rewarded with an extra life, or a more subtle weaving together of transitional game states and music as is seen in Taito Corporation's Space Invaders (1980), where a descending dorian scale pattern is repeated and gradually accelerates, the accelerando acting as the adaptive cue here, as the alien invaders are picked off one-by-one.



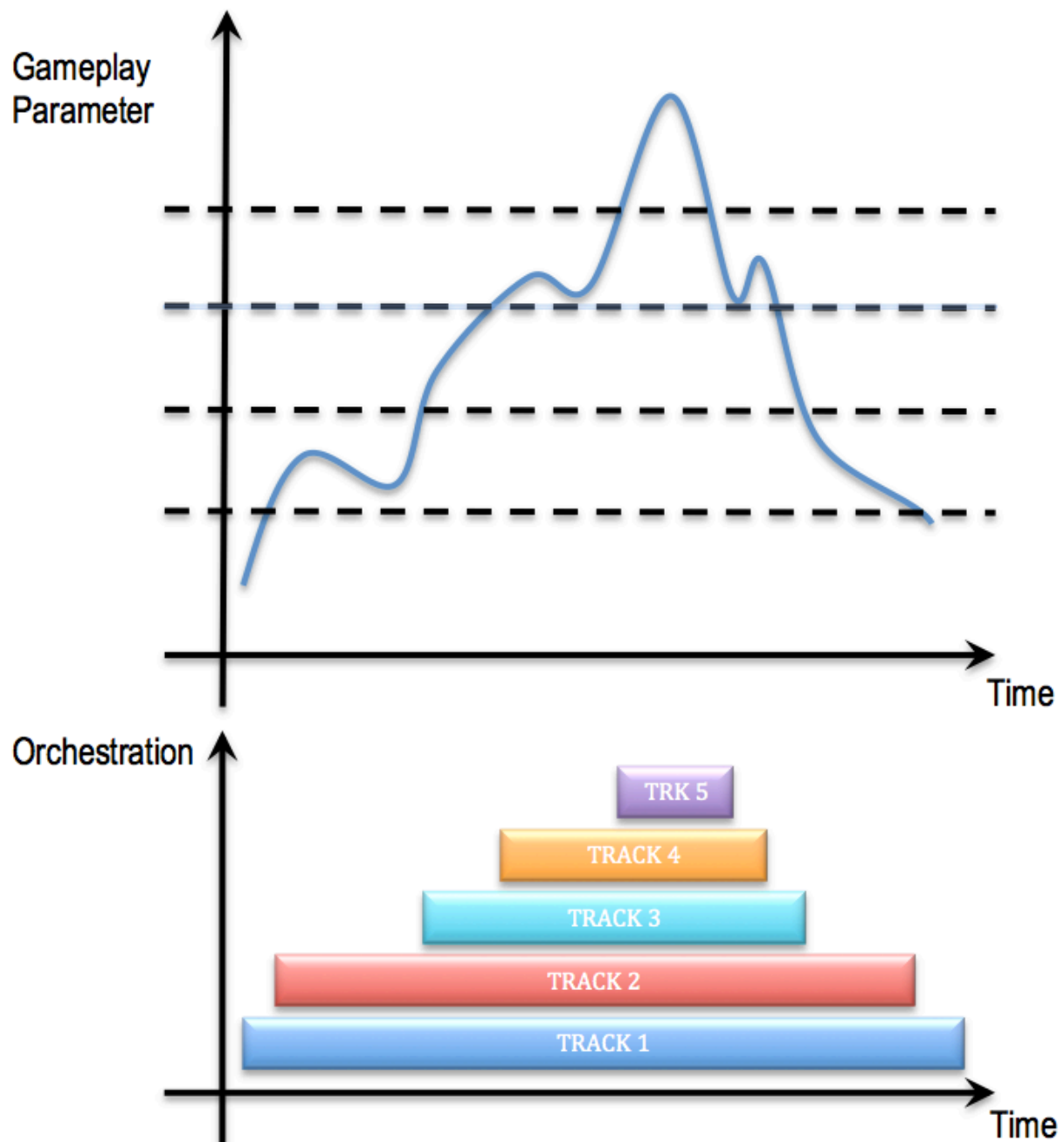
**Figure 1 – An event-driven music cue is triggered when a specific event occurs in a computer game. The cue is normally the result of a player action, which is detected by the game engine, and used to trigger a pre-written music cue.**

Horizontal resequencing, whose name derives from visualising the soundtrack extending along a horizontal timeline, uses similar gameplay markers to rearrange pre-recorded sequences of music in real time. Returning to Super Mario Bros, for example, the collection of a Super Star renders Mario invincible, and causes the main Mario soundtrack theme to be swapped for the faster-paced Starman invincibility theme for the duration of the power-up. The Mario intro fanfare is then used to segue back into the main Mario theme, which continues as the player progresses through the level.



**Figure 2 – In horizontal resequencing, the game engine monitors one or more gameplay variables in real time. When a pre-determined event or threshold value is reached, the game engine reorders the music sequence, in this case, substituting *Loop 2* for *Loop 1*.**

Vertical reorchestration similarly monitors the game state, but effectively remixes musical material in real-time, either dropping out elements of the soundtrack or building in intensity, depending upon the current gameplay state. The ‘vertical’ of the title relates to the discrete musical or sound layers that are stacked on top of one another as time progresses, a form of visualisation that should be familiar to anyone who has used a digital audio workstation (DAW) or sequencer, which typically has a timeline extending left to right along the bottom of the screen, with the individual tracks stacked vertically. The technique is used very effectively in the rhythm game, *Parappa the Rapper 2* (2001), which monitors gameplay and, if the player is struggling to keep time, it drops elements of the music out, before reintroducing them as the player’s performance starts to improve.



**Figure 3 – Vertical reorchestration monitors one or more gameplay variables in real time. When a pre-determined threshold value is reached, the music engine builds in intensity by introducing new pre-written musical components.**

Finally, procedural or algorithmic generation composes music ‘on-the-fly’ in response to current or anticipated game states. A good analogy here can be found in live jazz ensembles, which might consist of four or five musicians, each with slightly different roles to play within the group, and, although, as a collective, the band will have a rough idea of how each piece they play is going to sound, each of the musicians effectively composes in real time, creating

individual musical details as the piece develops. The key to successful jazz musicianship is for each musician to listen and adapt to what each of the other musicians are doing, so as to ensure that the performance is balanced and musical. Although there is no predetermined ‘script’ for the music, save, perhaps, for the loose framework provided by the chord changes and a sketch of the melodic head, good jazz musicians will pick up on cues from their fellow players, giving the music direction, and altogether generating a coherent output that exhibits the classical musical elements of tension and release.

This approach is less common than the other techniques discussed here, since to work effectively, it requires some form of artificial ‘musical intelligence’, and, ideally, some form of music performance engine to be included, both of which rob valuable memory space and processing power from the rest of the game. Nevertheless, a number of games have incorporated generative music as part of the gameplay. In Nintendo’s *Otocky* (1987), for example, the player controls a ship that fires round balls at enemy shapes and flying musical notes. With a simple two-note bass line, the player’s firing actions and in-game mechanics become the melody part, quantized in real time to the beat.

Typically, games use some combination of the above methods, sometimes to great effect. LucasArts’ iMuse system, for example, works beautifully in *Monkey Island 2* (LucasArts 1991), creating a rich and evocative evolving soundtrack, which, crucially, is subtle enough to be genuinely affective without drawing attention to itself.

Therein lies the crux of the problem here. For the video game composer, the structural methods that support adaptive music impose constraints on the way music is written to work with them. Composers often have to write game underscore blind to provide feedback on game-states which cannot be determined in advance, and must consider the different ways that one section of music might transition into another, perhaps using techniques such as tempo- or key-matching, that is, writing contrasting sequences in parallel at the same tempo and in the same key, to ensure that such transitions can be made if not seamlessly, at least without losing musical coherence. Compositionally, it is similar to a very complex and technologically-mediated game of *Musikalisches Würfelspiel* (Sweet 2014, p111), but for students to properly appreciate the subtleties and implications of this, they must first understand the structural edifices of adaptive music, and how these work, both individually and in combination.

The challenge with BitBox!, then, was to devise an experiential learning tool that could do that: one that would deconstruct the different elements of adaptive music and present them to students in a classroom environment in a way that

was accessible, and which allowed students to explore and manipulate the different aspects of the music in a controlled way. This initial proof-of-concept study focuses on the interface design for this learning tool. Its efficacy will be tested in a future cohort study planned to run during 2015 and 2016.

### **3. Design for musical play**

The approach to the development of BitBox! was, and continues to be, one of iterative design, an approach that has a long history of use in the development of human computer interfaces. This approach is particularly pertinent to the design of DMIs, which Orio et al describe as ‘a highly specialized field of HCI, dealing with such specific topics as simultaneous multiparametric control, timing and rhythm, and training.’ (2001)

Traditional acoustic instruments have layouts, interfaces and modes of interaction that are largely determined by their physical properties. A simple woodwind instrument, for example, requires some way for the player to excite a column of air so as to create a standing wave, and so, to provide airflow at one end, we might use a simple mouthpiece. The frequency of the standing wave, and hence the pitch of the instrument, will be determined by the length of the pipe. Since it is not particularly convenient to provide separate pipes for each note – although there are, of course exceptions to this, the pipe organ and the pan pipes being two examples – we might use a series of finger holes to vary the internal pressure within the pipe, effectively altering its length. Already, these elements, an end-blown mouthpiece connected to a small-bore cylindrical pipe with finger holes along its length, which have been dictated by the physical demands of tone production and modification, suggest a design that is very similar to that of the recorder.

Of course, there are still opportunities to refine or specialise the design. One might, for example, incorporate different types of single or double reed into the mouthpiece to alter the type of excitation and vary the tone of the instrument; or increase the bore and/or length of the instrument to change its tone and register, although this may also necessitate the addition of keys so as to open and close tone holes that are too large or too difficult to reach for fingers alone, and the end of the pipe could be flared to provide more natural amplification. These alterations in instrument design and interface all arise from the changing physical characteristics of the instrument and the constraints that these impose, and all point to the different members of the woodwind family.

Similarly, the player’s gestures and interactions with an acoustic instrument are determined largely by its physical qualities. In many instances the gestures are direct and immediate, and the performer’s actions, as is the case in the recorder,



play a direct role in exciting the acoustic mechanism, while in others, such as the piano, the player's input is mediated by a mechanical linkage. The relationship between gesture and acoustic event, however, typically follows a 'one-gesture-to-one-acoustic event' paradigm (Wessel & Wright, 2002).

DMIs, on the other hand, are not constrained in the same way. Consisting of a control interface, normally a set of sensors, which translate physical inputs from the performer to a set of control signals, which may be analogue or digital, the DMI isolates the front-end control interface of the instrument from the back-end tone generation and auditory output.

It is an approach that is similar to the drive-by-wire control systems used in many luxury cars (Zurawski, 2005, p29-2). Rather than use a traditional mechanical braking system, which uses pressure applied to the brake pedal to compress fluid in a hydraulic line and actuate the brake pistons to apply braking, a brake-by-wire system uses a group of sensors connected to the brake pedal to monitor its motion and the pressure applied by the driver's foot. These raw data are analysed by a processor, which attempts to interpret them and infer the driver's intent. The processor then generates an appropriate set of control signals, which it sends to the braking system, hopefully in the manner that the driver intended. The components of the brake-by-wire system are physically independent, and it is only by interpreting the meaning of the physical input that is sent to it that the processor can infer the action-intent of the driver and generate appropriate control signals.

Because of this physical independence between the different component parts of a DMI, the control gestures are essentially unconstrained (Hunt & Kirk, 2000). Anything that can be digitally captured and encoded can be used to generate control. Similarly, with no direct physical connection between the player's gestures and the sound produced, these gestures can be mapped to provide control over any of the control parameters, and with much more complex one-to-many; many-to-one, or many-to-many relationships between gesture and control.

Such flexibility in design capability raises interesting questions about its application in musical interfaces, and, since there is no single permitted set of options, how best to balance complexity and subtlety of control against user agency and the learnability of the instrument. Orio et al (2001) suggest a method for evaluating musical interface designs, and from this, the following assessment criteria were adopted:

- *Learnability* and the need to take into account the time needed to learn how to control a performance using a new controller. This may, in turn

influence the design itself, since designs which build on familiar musical gestures or which require minimal adaption of existing technique are likely to improve the learnability of a new device.

- *Explorability*, particularly of the range and number of different gestures or gesture nuances that can be applied.
- *Feature controllability*, taking into account the repeatability, and the accuracy and resolution of the interface as perceived by the user within the context of a particular musical task.
- *Timing controllability*, and the temporal precision at which the musician can control the performance.

#### 4. Gesture and Feedback

Much of the relationship between performer and instrument and between performer and listener is embodied in a series of gestures. As Goldstein notes:

*Music is a performing art, and part of the quality of the musical experience comes from the relationship between the player's physical technique and the sound that is produced. A listener can appreciate this connection visually (and viscerally) whether in a live concert or in the mind's eye while listening to a recorded performance. Our rich tradition of musical instruments has created a repertoire of gestures (bowing, blowing, banging, etc.) that are closely tied to familiar sounds. (1998)*

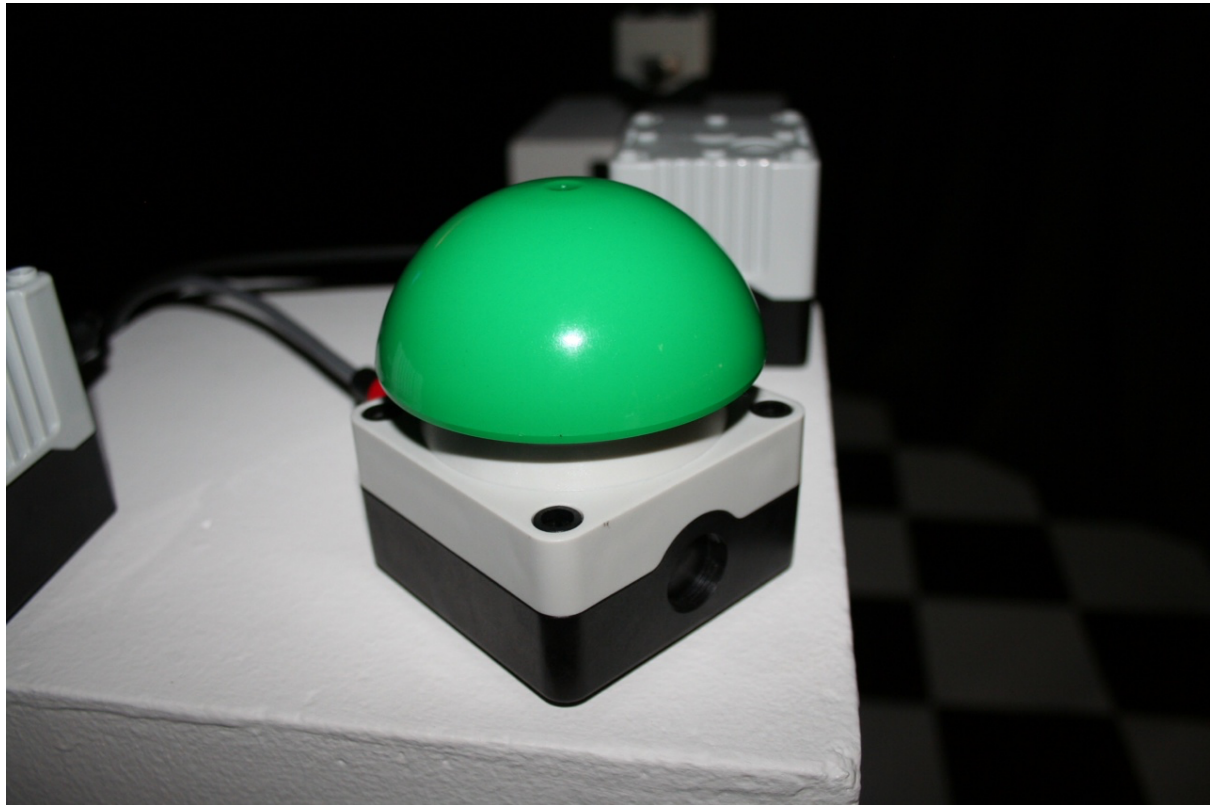
Indeed, this repertoire of musical gestures is, at least in part, well-understood and replicable even outside of the community of expert performance practitioners. One need only look at Rowan Atkinson's piano player or drummer mime routines (see, for example, Atkinson 2010) to see the subtlety of detail that can be communicated by musical gestures and how these can be subverted and understood even by non-players. In that respect, music gestural information can also be considered as a form of signal, some of which are necessary for the generation of sound, and some of which signal the emotional intent of the performer or other non-functional aspects of performance (Wanderley & Depalle, 2004).

Recent advances in the size, power and cost of microprocessors and sensors have now made it possible to capture and encode digitally spatial gestures and body movements relatively easily and use these to control DMIs. However, unlike graphical user interfaces (GUIs), whose key strength lies in making visible and, thus, through exploration, discoverable, the possible interactions a user might have with a system, gestural interfaces typically have to be learned, and often they require significant practice to refine and control. Donald Norman emphasises the fact that while gestural control systems offer interesting

potential for interaction and control, they are no different from any other form of interaction, and should provide well-defined modes of expression, a clear conceptual model of the way users interact with the system, and the consequences of their actions. (Norman, 2010). This contrasts, perhaps, with many recent gestural instruments, whose design is tailored to the needs of a particular performer, looking to translate their personal language of gesture into sound (Rovan & Hayward, 2000).

BitBox! was conceived to function as a teaching and learning tool for students keen to compose real-time adaptive music for interactive media, and, due in part to the role of music technology in contemporary and popular music composition, it cannot be assumed that students will have any practical experience of instrumental technique, nor any detailed knowledge of formal music theory. Further, while adopting a set of gestures from one class of musical instrument might improve learnability for students who are already skilled in applying that particular set of musical gestures, it could act as a barrier to those who are more familiar with other instruments or who have no previous experience. As such, there was an identifiable need to keep the gestural vocabulary simple, particularly for this initial prototype installation, and, as noted above, there was a clear imperative to employ gestures that resonated with the user's conceptual model of the instrument and their interaction with it.

Of the four different approaches to adaptive music, the event-driven cue is, perhaps, the most simple and direct, and can be expressed logically as 'if this happens, then play that'. In a game, the associated game event provides the inciting action and the music cue provides feedback to the player. For BitBox!, an industrial 22mm domed momentary contact switch was modified and a locking 6.35mm mono jack connector was added to its housing so as to allow easy connection to a central controller module, the idea being to provide an inviting – and shiny green buttons do hold a fascination for the inquisitive and mischievous – and tactile method of control input, the button press acting as a proxy for the game event. In the case of the BitBox! a short percussive break was selected so that the cue could be triggered on top of a backing track at any point in the music without breaking musical continuity or sounding discordant.



**Figure 4 – A 22mm momentary contact switch was wired to an Arduino Uno microprocessor to provide a means of triggering an event-driven music cue.**

It is not uncommon, particularly in the first-person shooter genre of games, for proximity to be used as a control driver for game music, with the soundtrack changing as the player moves between game zones, or closer to danger or an end-of-level boss. Proximity, then, seemed to provide a sensible basis for a gestural metaphor, since it provided a clear conceptual link between the DMI and its game music analogue, and had a clear music performance precursor in the Theremin.

Invented by Léon Theremin in 1919, the instrument uses near-field capacitance as the basis of tone control, with the player moving his or her hands in close proximity to two antennae, generating a constantly changing capacitance. The player's right hand controls pitch and the left amplitude. The gestures are natural and easy-to understand, but playing the Theremin is notoriously difficult, since pitch, in particular, is very sensitive to hand position, and, since hand capacitance approximately follows an inverse-square law, pitch sensitivity increases the closer the player's hands get to the antennae. Further, the instrument provides no tactile, only auditory feedback, which requires a very high degree of co-ordination between the player's aural and motor skills.

To address the issue of the sensitivity of nonlinear response, ultrasonic proximity detection was used in place of hand capacitance. Ultrasonic sensors are now well-established in noncontact presence, proximity, or distance measuring applications and are available as modular components that are easily incorporated into prototype circuits, and can be configured to provide linear output either digitally or by means of a continuously variable analogue voltage (Massa, 1999).

Maxsonar MB1000-EZ0 sensors (Maxbotix Inc., 2015) were selected for BitBox!, since these are relatively low cost; auto-calibrate; provide excellent range and beam width, and provide continuous range output. To continue the aesthetic design of the industrial contact switch, the sensors were mounted within plastic switch housings as shown in Figure 5 below. Each sensor required a ground connection, a +5V power connection and an analogue return line, and these were hard wired to a standard male XLR connector at the rear of the switch box so as to allow the sensors to be mounted and aligned according to the needs of the space and situation, with a 5  $\mu$ F smoothing capacitor connected in-line with the analogue output. These were tested with shielded XLR cables to a length of 10m without any noticeable degradation in signal quality.



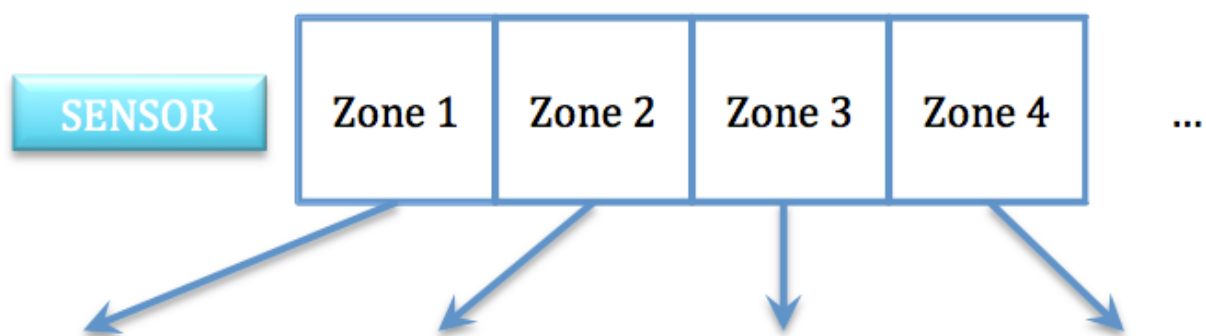
**Figure 5 – The ultrasonic sensors were mounted in industrial switch housings with a rear-mounted XLR socket providing connectivity to a remote microcontroller.**

To provide consistency of gesture across the remaining functions of BitBox!, that is, to control the horizontal resequencing, vertical reorchestration and procedural generation, four such sensor boxes were constructed. A central controller module was then constructed using a project box that housed an Arduino Uno microcontroller and a small breadboard, which distributed the +5V power signal, provided a common ground, and routed the analogue return signals from the sensors to the appropriate inputs on the Arduino. The housing was drilled to provide four female XLR sockets to provide wired connections to the remote ultrasonic sensors and a locking 6.35mm jack socket to provide a wired connection to the switch. Finally, a USB B socket was mounted to the case and wired to the USB connector on the Arduino to enable the control module to be connected to a host computer.

An Arduino sketch was written to sample the continuous analogue output streams from each ultrasonic sensor in sequence, with a 25ms delay between each to minimise the effects of crosstalk. The input data were converted into MIDI control messages, and transmitted via USB to a host computer running sequencing software. A looping backing track of around 3 minutes' duration,

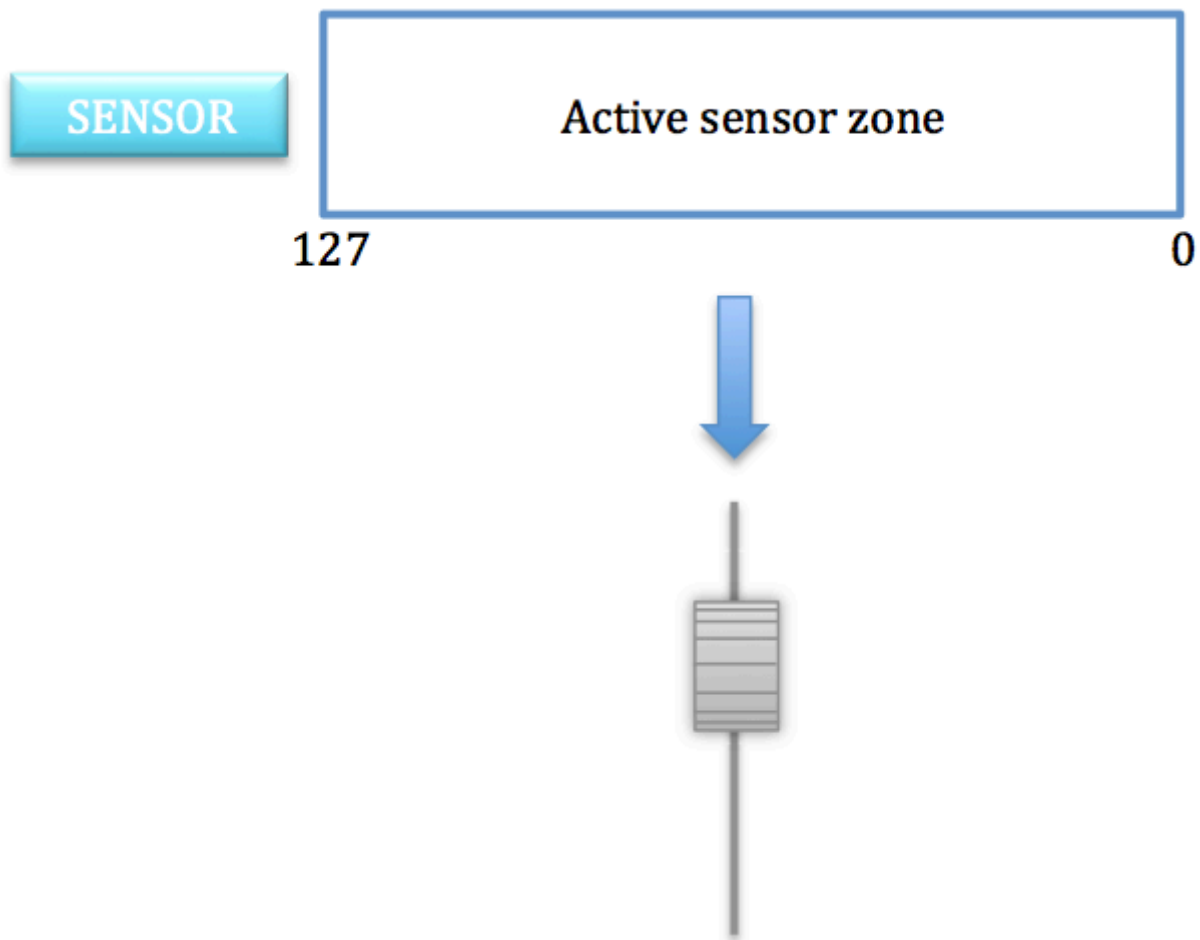
and which was intended to play continuously, was composed, and a number of additional sound layers and sequences, which would be controlled from the MIDI control data output from BitBox!, were written, and configured with custom scripts to control them.

For horizontal resequencing, the sensor data were spatially quantized into eight discrete zones, seven active positional zones and ‘off’, each around 10cm deep. These zones were then used to selectively solo one of seven tempo-matched and key-matched arpeggiated sequences which played continuously and synchronously in the sequencer, effectively allowing the user to compose new melodic lines by resequencing fragments of the longer arpeggiated lines.



**Figure 6 – The proximity of the player’s hand relative to the sensor generates continuous analogue data that are quantized to provide seven active spatial zones. These selectively solo precomposed arpeggiated sequences, allowing the user to compose by resequencing fragments of those sequences.**

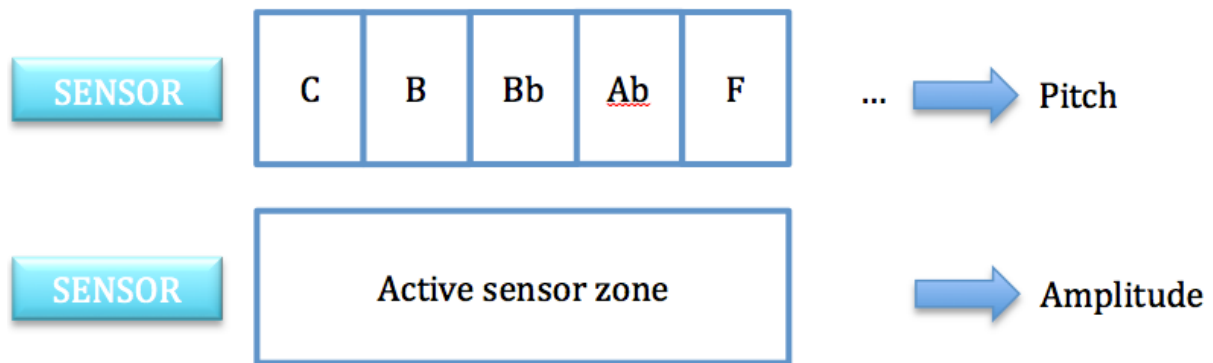
In BitBox! vertical reorchestration uses the continuous analogue data stream from a single ultrasonic sensor without spatial quantization, digitised and remapped to the range [0, 127] to control the channel level automation of a single audio track, gradually fading in an ambient layer in proportion to the proximity of the player’s hand to the sensor.



**Figure 7 – The proximity of the player’s hand relative to the sensor generates continuous analogue data that are linearly translated to the range [0, 127] and used to control channel level automation for a single ambient layer.**

Two ultrasonic sensors were used in combination to demonstrate the concept of procedurally-generated music, adopting the interface of the Theremin to translate the players’ hand gestures into pitch and amplitude control signals which drive a custom software simulation. In order to simplify the playing experience and ensure that players would be able to produce a melody line that was reasonably concordant with the backing tracks and the other interactive elements of the music, the pitch space, like that of the horizontal resequencing sensor, was quantized into 16 discrete zones so as to provide an ‘invisible keyboard’, with each zone mapped to successive notes of the blues scale. The amplitude sensor, like that for vertical reorchestration, was scaled to the range [0, 127] and sent as continuous controller data to alter the amplitude of the oscillator in the Theremin.





**Figure 8 – The demonstration of procedurally generated music uses two ultrasonic sensors and is implemented using a modified Theremin interface. The pitch data are quantized into zones, with each zone mapped to successive pitches of the blues scale, which controls the pitch of the Theremin’s oscillator. The amplitude data are linearly scaled to the range [0, 127] and used to generate continuous MIDI controller data to control the oscillator amplitude.**

## 5. Summary of results

The BitBox! interface was initially tested in a classroom with twenty-five undergraduate students drawn from BA (Hons) Creative Sound Production. This first testing round was used primarily to test the robustness and stability of the system. The system was run continuously for thirty-six hours, and monitored for system crashes, slowdowns and overheating. It was also tested with single and multiple users simultaneously to ensure that the system remained stable while all sensors were in use.

The full BitBox! installation was presented at Dare Protoplay 2014, an international games expo, which ran from 7<sup>th</sup> – 10<sup>th</sup> August in the city’s Caird Hall. The event attracted 13000 visitors. The main audience (approximately 42% of the total) comes from the 15 – 25 age group, providing a good representative sample of the target audience for BitBox! The instrument was enclosed in a 3 x 4 metre space, with the sensors mounted on wooden plinths placed centrally within the space and arranged to minimise sensor crosstalk. A real-time sound visualiser was used to generate synchronous imagery that was displayed on a 42” LCD screen to the rear of the space. The control module, the host computer, and an amplifier and speaker were hidden from view at the rear of the presentation space.



**Figure 9 – The BitBox! sensors were mounted on wooden plinths in the centre of the space and arranged to minimise crosstalk between sensors. A 42” LCD screen was mounted at the rear of the space to provide a real-time visualisation of the music.**

Sessions with BitBox! were arranged in small groups of between two and five users. Each session began with a brief explanation of the interface, its functions and its operation, and the concepts to which each sensor related, and a

demonstration of its use. Users were then encouraged to explore and experiment with the interface, and their interactions were observed during use. Following a period of free-experimentation, users were asked to provide feedback on their experience.

The sample group surveyed unanimously responded positively to BitBox!, reporting that it was easy to use. In particular, they noted that the hand gestures adopted as the main control mechanism felt natural, and were both easy to control and expressive, making it easy to control the musical output, even without any previous instrumental experience. This was corroborated by observation, where it was interesting to note that the majority of individuals, after a brief period of tentative exploration, very quickly adopted a confident posture, and moved their hands purposefully and expressively, mimicking the stance and some of the gestures of an orchestral conductor, suggesting that users very quickly felt a real sense of musical agency through their actions. Three participants, who had music therapy backgrounds, noted that, like the Soundbeam MIDI control system, the BitBox! had real potential as a device for enabling meaningful and significant self-expression for individuals with limited mobility or dexterity.

One interesting consequence of the simplicity of the gestural control system was that it facilitated multi-user interaction. Users reported that their experience was improved by using BitBox! with others, and that they felt confident and uninhibited using the system collaboratively, even with strangers. Again, this is backed up by observation, where it was noted that multi-user interaction was entered into enthusiastically, and that there was very little difference in the type and nature of multi-user interactions between known and unknown groupings. This suggests, perhaps, that a simple expressive music interface might have a role to play in overcoming performance anxiety issues in ensemble playing, where, early-stage musicians, particularly if improvising or inventing, feel constrained by their perceived lack of technical or expressive ability on their instrument, which often impacts on performance and sets up negative reinforcement of the idea that they are just ‘not much good’ at playing in groups.

Collectively, then, the simplicity of the gestures and the simple and direct control mechanisms seem to suggest that the BitBox! ranks highly in terms of learnability, with players developing confidence in expression and an ability to play with others very quickly.

Observations of the users also suggested that the interface encouraged exploration. Almost all users were observed experimenting with hand orientation and shape, again similar to the performance gestures a conductor

might use to direct an orchestra. When questioned, the participants noted that they were keen to explore the levels of control that they had with each of the different sensors and control methods. Again, this would perhaps suggest that the users develop confidence and sufficiency in the interface quickly, and that there is scope for adding additional nuanced control, which might increase the subtlety of expression for users as their familiarity with the interface grows. As noted above, however, such gestures must be chosen carefully to ensure a cognitive relationship between gesture and musical action, and to ensure consistency with the gestural vocabulary of the interface as a whole.

The feature controllability of the interface was generally good. Users responded very positively to the event-driven cue and to the procedural and vertical reorchestration cues, with the event-driven cue being, by far, the participants' favourite method of control because of its tactility. Users reported that they understood well each of the control methods and how they related to game music concepts, and felt that they were able to control these expressively. The horizontal resequencing, however, was less successful. Participants noted that they understood the concept as it was explained to them, but they were less clear about its application because they were not as able to hear the effect clearly through BitBox!, particularly when other participants were playing alongside them. As such, participants did not feel as if they were really in control of the music here and expressed a preference for a more distinctive form of auditory feedback so that they might better relate their actions to changes in the music.

Similarly, with the exception of the horizontal resequencing controller, all of the users responded that they felt in control of the music, and that their actions drove immediate change, suggesting that the controller sampling rate, and thus the timing controllability, was appropriate.

Overall, both the participant responses and observations suggested that the initial interface was engaging and effective, with many participants returning for repeated sessions, and clear evidence of participants experimenting with the capabilities of the system.

## **6. Conclusions and future work**

The first iteration of BitBox! was largely successful, with both direct observation and user-reporting of interactions suggesting that the interface was simple to learn, easy to use, expressive and encouraged exploration and experimentation. Users found the gestures natural, and understood the cognitive link between gesture, control and the underlying functional metaphors of the

whole system, specifically the deconstructed structural control methods for game music.

There were, however, two main areas highlighted for improvement or development. Firstly, there was the issue of the effectiveness, both in terms of explanatory function and as a music-making tool, of the interface's implementation of horizontal resequencing, and additional work is necessary to understand fully the true nature of the issue here. It may be that the effect is too subtle when experienced as just a single layer of a much more complex music mix, and that the horizontal resequencing control should act more directly on the overall structure of the music or backing track, rather than on just one track; it may be that the music sequences that are used for the resequencing component are too similar in their arrangement, making it difficult to identify change, or it may be that the method of quantised spatial interfacing should be refined to better connect the user's inputs to musical actions. These will be explored in future iterations of the interface.

Observations suggested that users quickly gained confidence with the interface, and quickly reached the limits of the system's gestural control, suggesting that there is scope to introduce an expanded set of gestures. In so doing, it is important to choose a gestural vocabulary that is coherent and consistent with the fundamental gestures of the system, and with that in mind, there is a useful analogy to be drawn with the role of an orchestral conductor. Both the conductor and the user here are at the centre of an experiential performance. Both exert agency to direct the outcome and use gestural control to shape the detail of the performance, and there is a definite argument for drawing on elements of the gestural canon from conducting and adapting them for Bitbox!

Braem & Bräm (2000) provide a very comprehensive description of the syntax and grammar of conducting gestures, with a particular focus on the gestures of the non-dominant hand, which provides some scope for development here.

Alongside this, however, must come a refinement of the control hardware, however, since the current ultrasonic sensing provides too coarse a level of accuracy to reliably detect anything other than proximity, and some form of optical sensing or image recognition to replace or augment ultrasonic sensing may prove useful.

In addition, participant feedback suggested that a degree of tactile feedback as well as auditory and visual feedback, was welcome, and increased the users' sense of control and interaction with the system, and some consideration should be given to how this might be incorporated into future refinements of the interface.

To be really effective as a tool for learning adaptive composition, BitBox! should be used as part of a practice-led overall strategy, with the DMI being used to explain and give experience of interactive compositional methods, before allowing students the opportunity to put these ideas into practice by composing their own adaptive music tracks. At present, BitBox! is still a bespoke controller, and although it is possible to compose new musical material for it, it is not a particularly user-friendly process, and work remains to be done here to create both a user composition framework that would allow the easy customisation of music material to quickly prototype new ideas, and the integration of the system into a properly structured programme of learning.

The next phase of the interface design of the project is underway and will be presented at Dundee Science Centre in August 2015 as part of its 'Music Mayhem – The Science of Sound' exhibition, which runs from 27<sup>th</sup> June until 25<sup>th</sup> October. This will be used primarily as a second iteration of interface testing. A first round of cohort testing is proposed for UAD's new academic session, which commences in September 2015.

## 7. References

Adorno T. W. & Eisler H. (1947) *Composing for the Films*. New York: Oxford University Press.

Atkinson, R. (2010) Rowan Atkinson Live - Star of Mr.Bean - Funny invisible drum. [video] Available at: [https://www.youtube.com/watch?v=A\\_kloG2Z7tU](https://www.youtube.com/watch?v=A_kloG2Z7tU) [Accessed 29 Jun. 2015].

Braem, P. & Bräm, T. (2000) 'A pilot study of the expressive gestures used by classical orchestra conductors', *The signs of language revisited: An anthology to honor Ursula Bellugi and Edward Klima*, ed. Karen Emmorey, Harlan L. Lane, pp. 127 – 148.

Brown, R. S. (1995) *Overtones and Undertones: Reading Film Music*. Berkeley: University of California Press.

Don Juan. (1926) [film] USA: Warner Bros. Pictures, directed by Alan Crosland.

Goldstein, M. (1998) 'Gestural Coherence and Musical Interaction Design', *Proceedings of the 1998 IEEE International Conference on Systems, Man and Cybernetics (SMC'98)* pp. 1076-1079.

Hunt, A. & Kirk, R. (2000) 'Mapping strategies for musical performance-trends in gestural control of music', *Trends in Gestural Control of Music*, M. Wanderley and M. Battier, Eds. Paris, France: Institut de Recherche et Coordination Acoustique Musique, pp. 231 – 258.

King G. & Krzywinska T. (2006) *Tomb Raiders and Space Invaders: Videogame Forms and Contexts*, London: I. B. Tauris.

Kolesnik, P. & Wanderley, M. (2004) 'Recognition, analysis and performance with expressive conducting gestures', *Proceedings of the International Computer Music Conference*, pp. 572-575.

Massa, D. P. (1999) Choosing an Ultrasonic Sensor for Proximity or Distance Measurement part 1: Acoustic considerations, *Sensors*, 16/2 (3). Available at <<http://www.sensormag.com/sensors/acoustic-ultrasound/choosing-ultrasonic-sensor-proximity-or-distance-measurement-825>> [Accessed 29 June 2015]

Maxbotix Inc., (2015) LV-MaxSonar®-EZTM Series. [online] Available at: [http://maxbotix.com/documents/LV-MaxSonar-EZ\\_Datasheet.pdf](http://maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf) [Accessed 29 Jun. 2015].

McAlpine, K., Bett, M. and Scanlan, J. (2009), 'Approaches to creating real-time adaptive music in interactive entertainment: a musical perspective', *35th International Conference: Audio for Games*. Audio Engineering Society. London, UK. February 2009.

Nielsen, J. (1993) 'Iterative User Interface Design', *IEEE Computer* Vol.26 no.11 pp. 32-41.

Norman, D. A. (2010), 'Natural User Interfaces Are Not Natural', *Interactions*, May + June 2010, pp. 6 – 10.

Orio, N., Schnell, N. & Wanderley, M. M. (2001), 'Input Devices for Musical Expression: Borrowing Tools from HCI', *Proceedings of the Workshop New Interfaces for Musical Expression* [Online]. Available at: <[http://recherche.ircam.fr/equipes/analyse-synthese/wanderle/Gestes/Externe/chi\\_workshop\\_Final\\_v2ALL.pdf](http://recherche.ircam.fr/equipes/analyse-synthese/wanderle/Gestes/Externe/chi_workshop_Final_v2ALL.pdf)> [Accessed 16 June 2015]

Parappa the Rapper 2. (2001), [disc]. Japan: Sony Computer Entertainment.

Quick, J. M., Atkinson, R. K. & Lin, L. (2012), 'Empirical Taxonomies of Gameplay Enjoyment: Personality and Video Game Preference', *Game-Based Learning*, Vol. 2 Issue 3, pp. 11 – 31.

Rovan, J. & Hayward, V. (2000), 'Typology of Tactile Sounds and their Synthesis in Gesture-Driven Computer Music Performance', *Trends in Gestural Control of Music*, Wanderley, M. and Battier, M. (eds). Editions IRCAM, Paris, 2000.

Schiffrin, L. (2011), *Music Composition for Film and Television*, Milwaukee: Hal Leonard Corporation.

Space Invaders. (1978). [arcade board]. Japan: Taito Corporation.

Steamboat Willie. (1928). [film] USA: Disney Bros Studios, directed by Walt Disney and Ub Iwerks.

Super Mario Bros. (1985). [cartridge]. Japan: Nintendon Corporation.

Sweet, M. (2014) *Writing Interactive Music for Computer Games*. Boston: Addison Wesley.

The Jazz Singer. (1926). [film] USA: Warner Bros. Pictures, directed by Alan Crosland.

Wanderley, M. M. & Depalle, P. (2004), 'Gestural Control of Sound Synthesis', *Proceedings of the IEEE*, Vol. 92 , Issue 4: 632 – 644.

Wessel, D. & Wright, M. (2002), 'Problems and Prospects for Intimate Musical Control of Computers', *Computer Music Journal*, Fall 2002, Vol. 26, No. 3: 11-22

Williams, D. (2005) 'Bridging the methodological divide in game research', *Simulation & Gaming* 36 (4): 447–463.

Zurawski, R. (2005), *The industrial communication technology handbook*. Boca Raton, Fla.: Taylor & Francis.



## **8. Contact details and biography**

Dr Kenneth B. McAlpine  
School of Arts, Media and Computer Games  
University of Abertay Dundee  
Bell Street  
Dundee  
DD1 1HG

Email: [k.mcalpine@abertay.ac.uk](mailto:k.mcalpine@abertay.ac.uk)  
Tel: +44(0)1382 308600  
Fax: +44(0)1382 308627

Dr Kenny McAlpine joined Abertay University as part of the team who developed the university's degree programmes in Computer Game Technology and Computer Arts in the late 1990s, where he leads teaching and research in sound production and music performance. He has a particular interest in video game music and real-time adaptive music composition and performance, and has developed game- and play-based and participatory soundtracks for theatre and heritage. Kenny performs as a jazz organist, and can often be found playing 8-bit covers of classic video game themes, about which he is currently writing a book for Oxford University Press.