NLN-live, an application for live nonlinear and interactive instrumental music

N.J. van Nispen tot Pannerden
HKU University of the Arts Utrecht, School of Music and Technology, The Netherlands
than.vannispen@hku.nl

ABSTRACT

NLN-live is a dynamic score system for nonlinear and interactive instrumental music performances, based on music technologies used in video games. The goal of NLN-live is to facilitate dynamically controlled collaborative performances that make live interactive video games music possible.

The principle of NLN-live is simple: every musician of an ensemble repeatedly plays two fragments of music that are presented on a screen, say X and Y, with the musical content for X and Y being variable and changing, for example, in relation to the interaction with a video game.

The continuously varying music fragments are controlled remotely by the NLN-live application, while the speed of these changes is controlled by the conductor.

NLN-live has successfully been used with single musicians, ensembles, as well as a symphonic orchestra.

1. INTRODUCTION

Nonlinear and interactive music are well known in, for example, certain video games, where the music in the video game adapts to the gameplay [3, 4]. Nonlinear music in games can be considered as a very broad range of music techniques and strategies in order to make the music correspond to the nonlinear and interactive nature of video games.

The two most frequently used techniques of nonlinear and interactive music in video games are vertical re-orchestration and horizontal re-sequencing, also known as 'the variable (open) form' (see Figure 1.) [3, 4].

In music history and concert contexts similar nonlinear music concepts are known, for example in compositions with an open form as well as compositions considered as aleatoric music such as Terry Riley's 1964 in C, Karlheinz Stockhausen's 1956 Klavierstück XI [1], Mozart's 1792 Musikalisches Würfelspiel and Jason Freeman's 2009 Piano Etudes [2]. In contrast to interactive game music these concert pieces normally do not have any interaction during the concert, but have other algorithms that realise the performance, such as randomness, chance or performers choice.

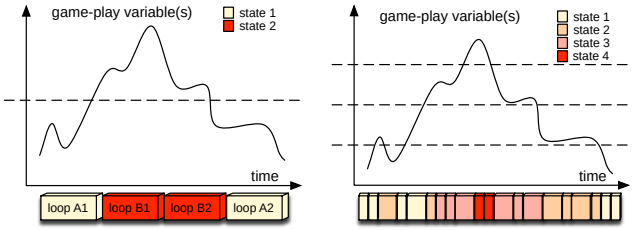

Figure 1. Two examples of horizontal re-sequencing. The shorter the loops the faster the response to the gameplay variable or interaction [4, 5].

Live instrumental music performances or concerts that feature the music of video games in an interactive form are extremely rare. This is mainly due to the lack of interactive music solutions for live instrumental performances and results in concerts with linearly arranged versions of (originally) nonlinear video game music (Games in Concert, Video Games Live, Indie Games Concert, King of Games, etc.). This music is often combined with (linear) video footage of the game, instead of actual video games being played, thus removing an important aspect of video games: live interaction. The aim of developing the NLN live application was to offer a solution for this live interactive instrumental music, for example in the context of video games in concert. NLN-live has been primarily based on the technique of horizontal re-sequencing (Figure 1.).

2. NLN-LIVE ARCHITECTURE

NLN-live is an open source Node.js web application based on horizontal re-sequencing (see Figure 1.) and consists of three different, interrelated, interfaces, a data-layer, the Dijkstra graph search algorithm and the pulse (see Figure 2.). All three interfaces are designed to fit on a tablet-screen. All interfaces are connected to the NLN live-server using network sockets.

2.1 Interfaces

2.1.1 Musician interface

The musician interface consists of a dropdown menu, to select an instrument, and two fragments of music that can be considered as encapsulated between repeat signs (see Figure 3.). This means that the musicians alternatingly play the top (X) and bottom (Y) fragments. Put otherwise: in Figure 3. X is the current music fragment, Y is the future fragment. If X has been played, Y becomes the current fragment and X the new future fragment. Thanks to

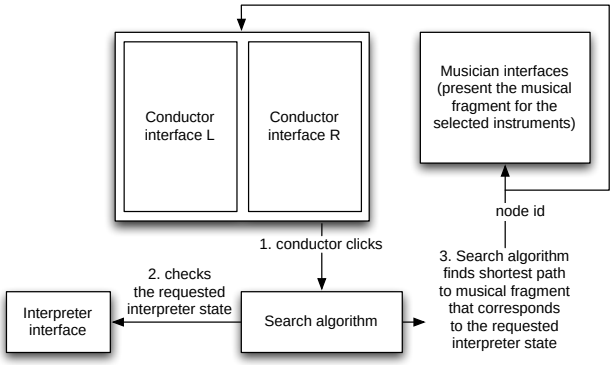

Figure 2. Basic architecture of NLN-live.

a coloured border around the current fragment (fragment X in Figure 3.) every musician knows which music fragment to play. The musical content of individual fragments changes once they have been played, while becoming the future fragment.

Musicians can always read ahead, since there is always a currently playing and future fragment.

The musicians could theoretically receive an endless sequence of music fragments and keep playing without turning a page, as their parts are controlled remotely and dynamically.

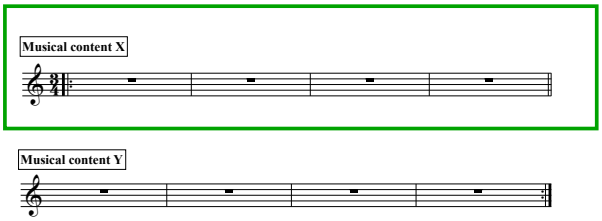

Figure 3. Two music fragments X, Y, encapsulated between repeat signs and with a border around the current music fragment.

The conductor performs the same role as in traditional performances and informs the musicians about the tempo and pulse of the music.

2.1.2 Conductor interface

The conductor has a similar interface as the musicians with two (a current and a future) music fragments being presented. Because of the size of the score, the music fragments are presented on a left and a right page on two separate tablets, or on a computer screen. It is the conductor, or the conductor-assistant [1] , who determines when the next music fragment should be presented to all musicians by tapping (or clicking) on the screen of one of the presented fragments. This tapping in the conductor interface (Figure 2.) can be considered as the pulse of the NLN-live application (see: section 2.4).

[1] Because the conductor might want to use both hands for conducting, and not for tapping or clicking, the conductor-assistant can do that for the conductor on a separate tablet or computer

2.1.3 Interpreter interface

The interpreter interface determines the variable to which the interactive music should respond. This interpretation can be connected to a separate algorithm, or application, such as a video game, but can just as well be a slider controlled by a music director. The value or state of the interpreter interface eventually determines which music fragments will be played.

On a linear slider with for example 3 values, a state of 2 would mean that music fragments that are member of group 2 are requested to be presented to the musicians and conductor. The (Dijkstra) search algorithm (see: section 2.3) finds the shortest path to music fragments that correspond to this requested state.

The linear slider of the interpreter can be considered as a state selector and a 2, 3 or multidimensional interface (see Figure 5.) could also be used, depending on the interactions and possible connections between music fragments.

2.2 Data layer

Every interactive music piece consists of music fragments with unique ids (nodes) stored in separate folders for every instrument and a database with all relations between these nodes (edges).

2.2.1 Music fragments

In the current version of NLN-live the music fragments for the conductor (score) and all musicians (parts) are presented as 1 to 5 bars .PNG images, prepared in music notation software Finale 2012 and exported in bulk as .PNG images using Finale Script (see an example in Figure 4.).

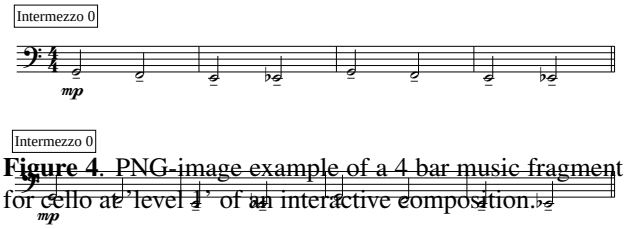

Figure 4. PNG-image example of a 4 bar music fragment for cello at 'level 1' of an interactive composition.

2.2.2 Nodes and edges

The database of music fragments (node ids) and their relations (edges) used in NLN-live was created using the graph visualisation environment arborjs' 'Halfviz' [2] (a dot-inspired, mostly declarative format for describing graph relationships).

When the conductor interface requests a new node (a new music fragment), the search algorithm finds a path to possible successors for the current node based on the current (requested) state of the interpreter interface and the currently playing node. The outcome of the search algorithm is the shortest path to a node matching the interpreters state and this outcome is distributed to the conductor interface as well as all musician interfaces. The conductor and musician interfaces use exactly the same node ids to present

[2] `http://arborjs.org/halfviz`

the music fragment for their corresponding (selected) instrument.

2.3 Graph search algorithm

To find the shortest path to a new node which matches the interpreters state an altered version of Dijkstra's algorithm is used. The regular version of Dijkstra's algorithm searches for the shortest path from node A to node B in a graph. The altered version of the algorithm searches for the shortest path from node A[1] to any node in a group of nodes B[] that match the interpreters state. It checks if any of the neighbouring nodes is in group B[], it then checks if any of their neighbours which were not checked yet are in group B[] and so on.

In Dijkstra's algorithm this calculation is done in a specific order, which will be the same every time the algorithm is run. This would mean that when the algorithm finds a path from A[1] to B[] it will always end up returning the same node as the solution, for example B[2], but it might be possible that more than one of the nodes in group B[] have a similar distance to A[1] as B[2]. For a musical application it doesn't make sense to always choose one of two possibilities if the other possibility is just as good. To solve this issue the algorithm continues, even after a first shortest path has been found for A[1] to a node in B[], to find all shortest paths to nodes in B[] and then randomly returns one of the found shortest paths.

Another adaptation to the original Dijkstra algorithm is the support for finding a path from node A[1] to itself, if the graph data would allow for it. In the NLN-live application this result would be considered as a direct neighbour.

2.4 The pulse

2.4.1 Timing

The conductor directs the musicians and sets the tempo of the piece.

By tapping (or clicking) in the conductor interface, after a music fragment has been played, the conductor decides 'the pulse' of the NLN-live application, or how fast a new fragment is presented. This speed is of course related to the conducting tempo, and the number of measures per musical fragment.

2.4.2 Latency

Conductors, as well as musicians, are used to reading ahead. This means that sight reading could actually be considered as sight reading of future (1 or 2 measures) musical material. These future measures, that need to be presented to the conductor and musicians, can be considered as the 'latency' of the musical system (the NLN-live application already presents music fragment Y in Figure 3., while music fragment X is being played, delaying the responsiveness to the interaction).

Early tests with professional musicians (from the Dutch Radio Philharmonic Orchestra) demonstrated that showing 2-4 measures in advance, in a moderate tempo, is enough to read ahead, although conductors and musicians mentioned that it would be more comfortable if they could see more measures in advance. Musicians also mentioned that a head start of 2 seconds is the bare minimum.

More experience and testing with the NLN-live application in live concerts is recommended to elaborate on the matter of sight reading and reading ahead in nonlinear music contexts.

These early tests led to the disclaimer for composers to write music that would consist of music fragments of at least 2-4 measures. In regard to the responsiveness and latency of the interactive music system, the music fragments were not allowed to be longer than 4-6 measures (the longer the music fragments are, the longer the, potential, response time to the interaction will be [5]). These compact music fragments also make it possible to easily fit the current and future music within the two staffs X and Y.

3. MUSICAL RESULTS

The NLN-live application has been tested with solo musicians, ensemble as well as a full symphonic orchestra.

3.1 *Musikalisches Würfelspiel*

Musikalisches Würfelspiel compositions by Mozart (K. 516f) and others have been prepared for the NLN-live application. Because of the short (1 bar) music fragments these become quite challenging pieces, when using the default musician interface which shows only 2 fragments.

3.2 *Mood Music*

At the occasion of the release of a new version of `http://ikcomponeer.nl` (an online interactive composition tool for children) an interactive piece has been written by Than van Nispen tot Pannerden. The composition was based on the affective circumplex model with a 2-dimensional valence-slider, or state-switcher, making it possible to select music considered 'neutral', 'content', 'happy', 'sad' or 'angry', with appropriate musical transitions (see Figure 5.) [6].

Unfortunately the concert, including this interactive composition, was cancelled due to 2013 cuts in funding by the Dutch government.

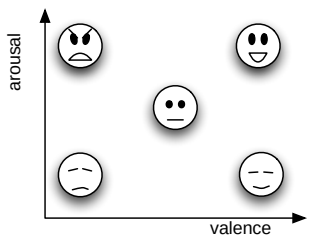

Figure 5. Simplified model of the affective circumplex model, usuable as a 2 dimensional slider for the interpreter interface.

3.3 *Space Invaders*

The Arnhem Philharmonic Orchestra (Het Gelders Orkest) premiered an interactive composition by Stan Koch and

Than van Nispen tot Pannerden at the Museumnacht 2013 while the video game Space Invaders [3] was played live. This new composition was based on the 4 note melody of this historic video game, which is considered the oldest interactive video game music [3].

Samsung kindly provided the 55 tablets for this live performance (see Figure 6.).

Figure 6. The Arnhem Philharmonic Orchestra, conducted by Ernst van Tiel, premieres live interactive music using the NLN-live web application on Samsung tablets, while the video game Space Invaders is being played live.

4. DISCUSSION

4.1 Notation difficulties in nonlinear score and parts

Because of the nondeterministic form of the music it is difficult to predict which music fragments will succeed one another. This has consequences for elements a composer would take in account in the composition, such as the notation of the dynamics (which in the discussed pieces are notated for every music fragment), instrument changes, use of mutes, harp pedals, key changes, but also to plan for breathing for brass and woodwinds.

4.2 Annotation

Musicians from different ensembles stated that they would appreciate some sort of annotation mode, in which they could make remarks and notes on top of the music fragments, similar to functionality in for example the ForScore and PiaScore applications [4]. Violin players are for example used to draw the bowing (up, down) in their parts. A workaround for this annotations could be to take over notes from a paper version, used during rehearsals.

4.3 Rehearsals

Because of the variable form of the music there is a chance that certain music fragments are not presented to the musicians during rehearsals, but are presented during the performance. To prevent the surprise of unfamiliar musical material it is recommended to have a linear version of all the music fragments for rehearsal practices.

This linear version of all the music fragments would ideally be available on paper for preparation and rehearsals. A linear version can also be presented on the musician interfaces when a linear sequence of the nodes is used in the graph data (1->2; 2->3; 3->...n).

4.4 Scores and parts as images

In the compositions discussed in section 3. the music fragments for the conductor and musicians were presented as .PNG images.

Modules such as vexflow and abc-notation javascript libraries have also been implemented and tested in the NLN-live musician interface, but either lacked some elementary notation elements required for the live performances, or required relatively large datafiles in regard to the PNG-files. However, for the use, or combination, of algorithmic music, the implementation of these modules could prove very valuable.

4.5 NLN-software

More information on NLN-live, latest versions of the open source NLN-live software, as well as installation instructions are available on: `www.nln-live.nl` and `https://github.com/supradeus/nln-live/`

Acknowledgments

The author would like to thank Muziekinstituut MultiMedia (MiMM), HKU University of the Arts Utrecht, Het Gelders Orkest, Ludwig ensemble, EYE Filmmuseum, Buma Stemra, Samsung, Ernst van Tiel, Loek Dikker, Bob Zimmerman, Jegor van Opdorp, Sebastiaan Donders, Stan Koch, Sander Huiberts, Rens Machielse and Marc Groenewegen.

5. REFERENCES

[1] W. de Ruiter, *Compositietechnieken in de twintigste eeuw*. De Toorts, Haarlem, 1993.

[2] J. Freeman, "Piano etudes." [Online]. Available: http://www.jasonfreeman.net/pianoetudes/

[3] K. Collins, *Game Sound*. MIT Press Books, 2008.

[4] K. B. McAlpine, M. Bett, and J. Scanlan, "Approaches to creating real-time adaptive music in interactive entertainment: A musical perspective," in *The Proceedings of the AES 35th INTERNATIONAL CONFERENCE*. AES Royal Academy of Engineering, 2009.

[5] T. van Nispen tot Pannerden, "The nln-player: A system for nonlinear music in games," *Proceedings of the International Computer Music Conference 2011, University of Huddersfield, UK*, 2011.

[6] J. Russell, "A circumplex model of affect." *Journal of Personality and Social Psychology*, vol. 39, pp. 1161–1178, 1980.

[3] Space Invaders (1978). Taito Corporation, Midway

[4] `http://forscore.co` resp. `http://piascore.com`