ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF ARTS AND SCIENCES

ONLINE GAMING AS A MODEL FOR NETWORKED INTRA-ACTIVE MUSIC SPACES

M.A. FINAL PROJECT

ÖMER CEM ÇAKMAK

Dr. Erol Üçer Centre for Advanced Studies in Music (MIAM)

Music Program

Final Project Advisor: Dr. Anıl ÇAMCI

MAY 2015

İSTANBUL TEKNİK ÜNİVERTESİ ★ SOSYAL BİLİMLER ENSTİTÜSÜ

AĞ İLETİŞİMLİ İNTRA-AKTİF MÜZİK MEKANLARINA ÇEVRİMİÇİ OYUN MODELLERİ

YÜKSEK LİSANS BİTİRME PROJESİ

ÖMER CEM ÇAKMAK
(409131027)

Dr. Erol Üçer Müzik İleri Araştırmalar Merkezi (MIAM)

Müzik Programı

Bitirme Projesi Danışmanı: Dr. Anıl ÇAMCI

MAYIS 2015

Ömer Cem ÇAKMAK, an **M.A.** student of ITU **Graduate School of Arts and Social Sciences** student ID **409131027** successfully defended the **final project** entitled **"ONLINE GAMING AS A MODEL FOR NETWORKED INTRA-ACTIVE MUSIC SPACES"** which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Dr. Anıl ÇAMCI** Istanbul Technical University	....
Jury Members :	**Dr. Reuben DE LAUTOUR** Istanbul Technical University	....
	Dr. Can KARADOĞAN Istanbul Technical University	....

Date of Submission : 4 May 2015

Date of Defense : **27 May 2015**

TABLE OF CONTENTS

Page

ABBREVIATIONS

DAW	**:** Digital Audio Workstation
FPS	**:** First Person Shooter
GCSM	**:** Generalized Client Server Model
IMN	**:** Interconnected Musical Networks
IP	**:** Internet Protocol
LAN	**:** Local Area Network
MDA	**:** Mechanics, Dynamics, Aesthetics
MIDI	**:** Musical Instrument Digital Interface
MMORPG	**:** Massively Multiplayer Online Role Playing Game
OSC	**:** Open Sound Control
Pd	**:** Puredata
TCP	**:** Transmission Control Protocol
UDP	**:** User Datagram Protocol
VE	**:** Virtual Environment
WAN	**:** Wide Area Network

LIST OF FIGURES

Page

ONLINE GAMING AS A MODEL

FOR NETWORKED INTRA-ACTIVE MUSIC SPACES

SUMMARY

This graduation project investigates the methods, challenges and possibilities of performing collaborative -or cooperative- music through networks, inspired by the dynamics and mechanics of online multiplayer games. The research concentrates on the following topics:

- Examination of music instruments in the digital world, the attempts to categorize new devices for music, how we define acoustic and electronic instruments, and how they separate from ordinary tools.
- An investigation of networked music, the idiomatic properties and challenges of network use in performing. Different types of network arrangements, past and recent examples of networked music systems, of which some include gaming methods.
- A brief study of games, methods to analyze game mechanics and user interactivity. Ontological properties of real and virtual environments, and behaviors of online gamers while playing in these environments. Game engines and their use in network music applications.
- Furthermore, supporting the conducted research, a game-like network music environment is created using available open source software development platforms, frameworks and libraries. A discussion on how it relates to the previous research and the general user response will be evaluated.

The project's final aim is to propose new methods of music making on networks and suggest the development of new software for networked music analogous to online gaming.

AĞ İLETİŞİMLİ İNTRA-AKTİF MÜZİK MEKANLARINA

ÇEVRİMİÇİ OYUN MODELLERİ

ÖZET

Bu bitirme projesi sanal ağlar üzerinden gerçekleştirilen müzik performanslarının yöntemlerini, zorluklarını ve olasılıklarını incelemekte, ve çevrimiçi oyunların modellerini uygulamayı tavsiye etmektedir. Değindiği noktalar sırasıyla yeni dijital müzik enstrümanlarını tanımlama ve kategorize etme yöntemleri, ağ iletişimli müzik sistemlerinin tanımı, kendine has özellikleri, projeyi yakından ilgilendiren örnekleri, genel anlamda oyunların kuruluş ve işleyiş şekilleri, sanal çevrelerin ontolojik tanımlaması ve oyun motorlarının ağ iletişimli müziklere nasıl altyapı oluşturabileceğidir. Son olarak bu araştırmalara paralel bir şekilde tasarlanmış Monad isimli beraber çevrimiçi müzik yapımını teşvik eden, oyunumsu, intra-aktif ve görsel bir program geliştirilmiştir. Proje nihai olarak internetin dahil edildiği müzikal işbirliği ve performans araçlarına çevrimiçi oyun kültürünün birtakım özelliklerini aktararak daha verimli ve özgün müzik enstrümanlarının üretilebileceğini ileri sürmektedir.

1. INTRODUCTION

How we experience music as well as other media today has little in common with the past. Throughout history, music could only exist when performed at a specific location, in a specific time. With the arriving technologies to contain a musical performance in a plastic medium and replay it as it was, music became detachable from a specific time and location, while preserving its cultural origin and background. 'Materializing' music this way enabled broad distribution of sound in tangible form. But as early as 1934, French philosopher and poet Paul Valéry predicted,

> Just as water, gas, and electricity are brought into our houses from far off to satisfy our needs in response to a minimal effort, so we shall be supplied with visual or auditory images, which will appear and disappear at a simple movement of the hand, hardly more than a sign.

Especially since the widespread establishment of digital technologies in the 1990s, we are in fact being constantly supplied by visual and auditory images, and making them appear and disappear is getting easier every day. Music in this situation, has transformed into information stored, sold, shared and streamed on computer networks. Now, it can be rapidly reproduced without any losses; similar works can be found and/or recommended by machine learning mechanisms or other users. The Internet has proved to be an effective and popular medium for music 'consumption', while the same cannot be said for real-time musical collaboration. To this day, making music 'together' is still widely regarded as a live and location-specific activity. Although the act of making music through networks has been practically synchronous with the development of network technologies throughout the 20th century, this effort never managed to break out of the 'niche' categorization.

Music as digital information, communicated between programs or between computers, has a highly flexible form. As stated above, we can 'contain' music in different types of media for some time, but especially with the rise of tape as a medium, a new surge of composers have utilized the electroacoustic domain as an expressive and artistic field. Tape offered a tangible material to cut, paste, speed up/down, record and re-record. Such an expanded control over sound fulfilled the needs of electronic music composers of the era, many of them coming from the serialist movement and extending the technique over to the electronic medium. As a result, electroacoustic music pieces became substantially 'frozen' (as opposed to 'live') and the composer could hold about every aspect under control, a highly fixed form of expression (Gresham-Lancaster, 1998). One would have expected from the digital medium, offering an even more detailed (i.e. sample based) control over sound, to continue pushing towards the extremes of precision in music. But in some cases, musicians started to take advantage of this domain to produce unintentional or causal sounds, even 'failures' as seen with the peak of the glitch movement (Cascone, 2000). Still, the effects of technological advancements on music have continuously given the composer increased autonomy as well as gradually changing his/her role. By the time digital audio workstations (DAWs) came to be widespread commercial products, the need for a music studio, let alone an accompanying musician, turned out to be non-essential. Musicianship, once a highly collective and collaborative activity, has conversely transformed into the opposite: an individual, isolated, anti-social occupation (Makelberge, 2012). This situation of autonomy and individualization is undergoing a radical change with the Internet becoming a common, fast and efficient way of connecting with one another. The cyberspace, comprised of vast amounts of information, resonates sound in its own way and offers its many inhabitants to collaborate, cooperate and collectively create either asynchronously or (almost) real-time. Whether this means sampling from other artists found online or forming live performance tools where users influence, participate, and develop each other's music in real-time (Weinberg, 2003), idiosyncratic properties of the net offer musicians to produce new and original musics for its own unique environment.

Today, we observe the possibility of bringing the 'play' aspect back in the center of making music, constructing open forms and letting the participants, computers or even the audience finish the piece. Electronic music, linked with other new media, is becoming a 'living' cultural expression more than ever in its history, leading to new and more comprehensive naming such as *sonic art* or *sound art*. The medium in which the sound information is captured, once highly rigid (i.e. wax cylinder), has become extremely flexible within the digital environment in terms of distribution and manipulation. In other words, music can now live, evolve and change within a synthetic medium in unforeseeable ways. This virtual space inherently calls for virtual (i.e. software-based) instruments, and thus synthetic music. Since the tools, the methods, and the attitude towards composing for this space will need to differ from previous traditional ways, the music should not be expected to produce similar results, although at times they just might. In order to realize a music idiomatic to this virtual environment, we need to address the characteristics of suitable tools and instruments, and where they are positioned among other musical devices.

Furthermore, this paper puts forward online gaming as a model for networked music performance. Multiplayer network games, unlike network music, have proved to be extremely popular, transforming into multi-billion dollar businesses (part due to online gambling) as well as attracting artists and independent developers as a creative platform. In 2014, U.S. sales for interactive software entertainment companies reached 21.5 billion dollars, reported by the Entertainment Software Association. Compared to the 16.7 billion dollar public revenue declared by the Motion Picture Association of America for the same year, we can clearly observe the presence of video games as an important choice of entertainment and time investment.

Popular examples such as Massively Multiplayer Online Role-Playing Games (MMORPGs) demonstrate highly efficient use of teamwork, collaboration, competition, and even spectatorship on networks. These densely populated games such as Second Life or World of Warcraft have transformed mainstream gaming into a primarily social platform of interactive multimedia. Online gaming, now considered a ‘sport’, has successfully used the methods of forming communities on the Internet and making impressive use of interaction between users logging in from around the globe. For these reasons, both the network infrastructure and user interaction associated with online games can set an example for future networked music environments. There is a strong overlap between the element of unpredictability in game dynamics leading to ‘fun’, and the uncertainties of network music systems that inflict different aesthetical results. In line with these thoughts, a networked music software named Monad accompanies this research, which seeks to realize these concepts of game design, communication and musical collaboration within a virtual musical environment.

2. TOOLS, DEVICES AND INSTRUMENTS FOR NEW MUSICS

Artistic works focusing on sound have branched off towards many different categories, ranging from acousmatic compositions to sound installations and computer games. Since the spaces where we are exposed to musical expression have diversified, so did our relationship with new musical tools. In line with this change, we can observe an effort to categorize the broad group of musical devices with varying functionalities and purposes (Pressing, 1999; Wanderley, 2002; Birnbaum et al., 2005; Magnusson, 2010). By establishing ways to define and evaluate emerging musical devices, the authors seek to find common patterns between them and help design new ones. Visual approaches to represent the new and diverse tools for music are suggested. The dimension space analyses examine interactive tools, programs and installations in both theoretical and practical characteristics (Birnbaum et al., 2005). While these new devices stimulate a need to define and articulate what sets them apart or makes them alike, there is a much finer line separating acoustic and digital instruments, which does not require as careful an examination to identify. It is evident that computer-oriented instruments are unalike traditional ones, but by addressing their epistemic nature we can help further develop digital devices as instruments aimed at networked music applications.

Traditional music instruments provide a physical extension to the performer to channel through their talent and expressiveness. The performer-instrument relationship is clearly embodied. But the generalized and undefined nature of digital instruments programmed and re-programmed to perfection does not provide a continuous extension from the body. Instead, they are external devices, separate from the individual, and the information they provide is available to interpret, thus hermeneutic (Magnusson, 2010). In essence, traditional instruments are tools as well, working under specific mechanical principles, but are seldom categorized as such. Since they operate under creative, technical and unique actions, a professional

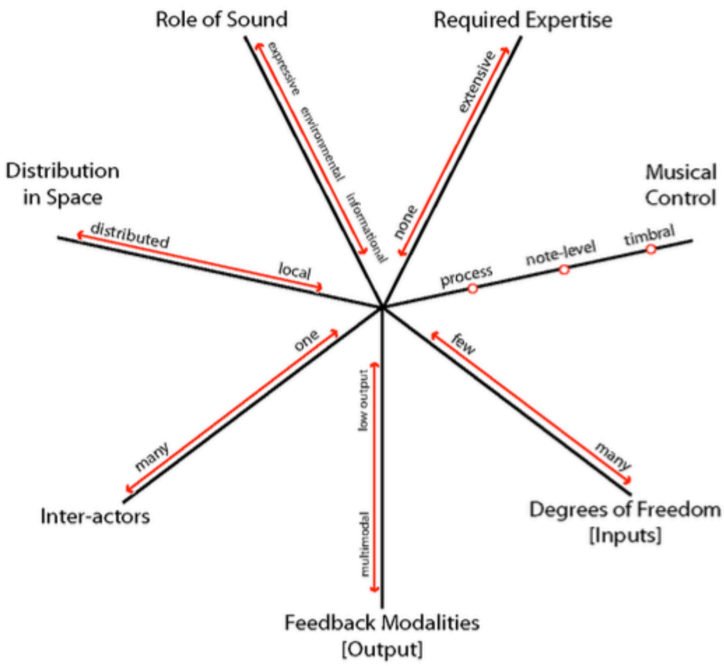

Figure 2.1 : Birnbaum's 7-axis Dimension Space for musical devices (Brinbaum et al, 2005)

instrument is often highly specialized. For this reason among others, some new electronic instruments (i.e. synthesizers) have adopted traditional interfaces (i.e. piano keys) to perform with. Tanaka (2006) explains the tool vs. instrument separation is mainly the cause of practical concerns while using a tool (which has to function for a service and can be improved upon) and the aesthetic considerations of an instrument, which needs to sound in a certain quality, taking on a character embodying limitations and imperfections. Thus, the way a musical instrument sets itself apart from ordinary tools is achieved by displaying an implicit personality.

So would this tool vs. instrument situation imply a lack of personality in digital music devices since they are mostly referred to as tools? Not necessarily, but the notion of computer as a musical instrument is not widely internalized by even computer musicians themselves (Wesel and Wright, 2002). It is true that computers are highly general devices and can be used for a wide range of purposes outside of music. Computer hardware itself represents nothing musically, but combined with appropriate software, it can produce original results. Even as a tool for creativity, computers automate human work, and this labor saving quality is their original design objective in the first place. What sets them apart from any ordinary industrial machine is that its meta-properties generate ideas and support thinking as well as the execution of these ideas, or tasks.

This issue with the lack of personality, yet possession of capacity with new tools for music led some composers, Luciano Berio in this case, to contemplate:

> […] It is easy and superfluous to produce new sounds that are not the product of musical thinking, just like it is easy today to develop and 'improve' electronic music technologies when they are disconnected from a deep and realistic musical context. With or without new tools and technologies, electronic music as a means for musical thinking reached a dead end. Moreover, the new tools detached it even further from the global and comprehensive idée of music making which is perceived not only by its technical, historical, and expressive terms, but in contemporary and social terms as well. (1983)

While one could assert Berio's claims are valid up to a certain degree for highly formalized electronic genres such as the acousmatic tradition, few would agree electronic music has been infertile in contributing to musical thinking since 1983. Furthermore, new tools not only extended the capabilities of electronic music making, but also gave birth to many original styles and characteristics. With these new devices, the idea of composing music itself has also been transformed. In that sense, Berio might have been correct to declare that the "idée of music making" has expired, since making music has been largely redefined in the years to follow, as well as the role of the composer.

The efforts to turn the computer into a sophisticated and expressive instrument mainly deal with the physical interaction with the device. Since all traditional instruments exert some sort of gestural involvement from their players, designing gestural interfaces as inputs has been a popular research area to modify digital tools into instruments. The ways we communicate with computers as instruments extend from simple and common keyboard-mouse inputs to highly customized sensors and tracking devices (Bongers et al., 1998). This directly corresponds with the expressive property of the tool, in most cases adding value to it as a performance instrument. But, in all examples of these interfaces as expressive inputs, what they do and how they function is only a matter of trans-domain mapping mostly done according to personal preferences. While this could improve the flexibility of the instrument and display its customizable properties, there is also the chance of losing its wholeness and drawing the attention towards the mapping techniques rather than a holistic statement.

The instruments of networked music systems present an extension of digital instrument characteristics discussed above. While designing expressive tools for virtual environments, we must consider different aspects such as its generality, depth of exploration, expressive limitations, and required skills. Thus, the built software will hold certain qualities and will occupy an epistemic dimension space as a musical device. Looking closer to network music tools among these new devices, different types of net music have been further categorized in relationship with each other. Föllmer (2005) positions network music types in a three dimensional space: network interplay, openness/interactivity and complexity/flexibility. Within this space, we can observe certain clusters that reflect similar structures and purposes. One cluster represents exchanging and archiving music products without affecting the production process, but this activity contributes to evolution of music in a larger time span nonetheless. Another group brings together toy-like music tools, easily accessible and enjoyable by everyone, providing limited and straightforward results. One of the problems with creating digital music devices is the choice between building an accessible one, which all net users can 'play' with, or an instrument that requires mastery over time. The latter type, concerning this paper, seeks for new modes of interaction and tries to increase the complexity in sonic results. But, the net music group related to performance is held separately from the instrument cluster. The author explains the challenge of bringing in audience participation without removing players with expertise from the setting. The effort to keep the audience in an interactive mode while performing elaborate instruments is still an ongoing issue. But, this project will further discuss ways to demonstrate live activity and enhance the audience-performer relationship in the following chapters.

3. MUSIC AND NETWORKS

3.1 What Are Networks?

Interrelations of linked devices form a network. These devices, called nodes, can be computers, printers, or (with the Internet of Things) any ordinary looking object that is able to send or receive digital information. Networks allow the transmission of all digital data such as code, images, sound, text and many more. This transmission is handled by a set of rules operating in a logical procedure, called protocols. Computer network protocols work on seven hierarchical layers standardized by International Standards Organization called Open Systems Interconnection (Zimmerman, 1980). The nodes in a computer network all have unique names, or addresses, to introduce, recognize and locate each other. Internet users, for example, connect using the Internet Protocol (IP), and therefore all have IP addresses. Open Sound Control (OSC) and Musical Instrument Digital Interface (MIDI) protocols are especially popular among network musicians, and most examples of networked music utilize either one or the other. Networks can be classified in two general categories: local area networks (LANs), and wide area networks (WANs) (Roads, 1996). As their names indicate, they are specified in relation to the geographical area they cover. LANs can be setup by independent users or institutions, and can include from two to hundreds of nodes. WANs on the other hand enable transmission using more sophisticated infrastructures such as telephone lines, and can interlink thousands or more devices together. Whether there are two computers in a network or thousands, all network organizations have common features such as recognizing each other (as well as themselves), connecting with one another, and knowing what protocols each machine is using (Noble, 2009).

In order to communicate with computers across vast distances, a common practice is to use a modem, which is short for modulator-demodulator. Modems convert (or modulate) the digital stream of information in one computer to an electrical signal (high-frequency audio) and send it through telephone lines. The receiving modem then demodulates the signal to feed the original digital information to the computer, enabling a dialogue between connected computers. But unlike human dialogue, modems send and receive transmission simultaneously, the rate of it depending on the modem capacity. There are many different solutions to interconnectivity problems, and the network musician should carefully decide on the technology, required budget and the performance of the system he/she decides to use in an ensemble.

3.2 Networking Music

Networked music relies on telecommunication systems. Due to this dependence, technology is an inherited and internalized feature. The performance limitations and specifications are strictly linked with the available technology, and the types of works vary with the types of transmission tools in use. For example, the earliest known instrument utilizing transmission, the Telharmonium (developed in 1897) transmitted electrical signals over telephone wires to a receiving end with primitive loudspeakers (paper cones placed on telephone receivers). Other conceptual works such as John Cage's 1951 piece "Imaginary Landscape No. 4" (for 12 radios) also makes use of transmission technology, but computer networks form the larger part of networked music performances after the increase in available digital devices throughout the 80s and 90s; along with the establishment of the Internet, computer networks need not be local anymore. This led to a more specific naming and definition, telematic music, as Oliveros, Weaver, et al (2007) elaborate, is "music performed live and simultaneously across geographic locations via the internet". For musicians, making use of the Internet as a musical communications tool implies new kinds of interactions (e.g. transporting sound and/or information of sound over larger distances) and new kinds of limitations (e.g. that of bandwidth). Whatever the advantages and disadvantages may be, telematic music systems and their performance are defined by the technology they are built upon, and the transmission medium itself, with its defects and limitations, becomes a statement (or one of the statements) of the artwork. Dealing with the network properties "as they are" provides the performers to play on and with the networks, not just in terms of connecting two sites together, but also by engaging in the common space it provides

(Caceres, 2008).

Before submitting to the concrete facts of telematic music and utilizing them for artistic expression, we should first demonstrate its inevitable physical limitations. When two or more musicians collaborate live, sounds from each performer need to be played within a window of 40 milliseconds in order to be perceived as simultaneous to the listener. This is called the precedence (also referred to as Haas) effect in psychoacoustics, and in the case of time differences greater than this window; we are able to separate the sounds as two distinct events. Barbosa (2003) illustrates a peer-to-peer connection between two locations on opposite sides of the globe with the length of half the Earth's perimeter, about 20004.5 kilometers. With a flawless data transfer rate at 300.000 km/s, the speed of light, the bidirectional delay time would still be around 133 ms, over three times larger than the Haas window. There is a ubiquitous boundary for network transmission due to the fundamental physical laws of our universe, and as long as the space-time continuum remains intact, network technologies will keep instigating delays that exceed the maximum allowed amount for performing traditionally live music.

New pieces of networked music parallel new technology of the time: from the transistor radio to digital logic circuits, and onwards to the Internet. While the common market for these products is naturally concerned with their functionality, artists and researchers mainly deal with their constraints and cultural suggestions (Rohrhuber, 2007). Often times, composers take advantage of these technological innovations in ways that are unintended by its original developers. Graham-Lancaster (1998) recalls the instance when an engineer observing David Tudor's set-up noticed a device of his own making, but patched 'wrong' in the system. The artist explained that using it in this configuration provided a remarkable feedback, which he wanted to keep. As a result, "the engineer was dumbfounded, the audience rewarded".

3.3 Composing for Networks

So why is telematic music important? Although communicating lies in the heart of it as in every other music, there are other aspects to think about. At this point we should also consider the concept of telepresence, where the performers feel -and eventually behave- as if they are present in a remote location. This location, in fact, need not be physical. As observed with online gamers in the previous section, it can be a virtual space inhabited by users connected from various places around the globe, where they can play, create and collaborate. The obvious implication of this is that the geographic locations will be eradicated, however, this does not fill in the whole picture; transmission of data over the Internet pans out with latencies and quality losses.

> Given this kind of situation, a traditional musician could not expect to perform music as if he is normally accustomed to. 'But the timing is strange,' the musician might say, 'how can we play our music this way, it's not going to work.' My reply always was that the musician could not expect to impose his music unaltered onto a new time/space domain. The technology, contrary to what is often advertised, is not transparent. While the typical reaction of a musician was to ask if the technology could be improved to eliminate latency, my response as composer was not to re-program network algorithms, but to write music for the given situation. To me it was somehow appropriate that any given music could not simply be transplanted and successfully performed on a network infrastructure (Tanaka, 2006, p. 273).

Composing music for particular locations is by no means a new attitude in music; European composers during the Medieval Ages would write music specifically for large reverberant churches, and used the properties of the performance space to conceal secular melodies added in the works. Bebop musicians performing in small club spaces of the time, for example, would take advantage of the sharp response of the small room by playing short notes in blazing speeds to match the environment. As in every other type of performed music, telematic music too, finds its own personality when it makes use of its environment's properties. But of course, network music systems come in more than one shape, form and purpose, and we can see its utilization in many other areas.

Often times, new technology first encounters old habits; much like in the case of Theremin's arrival, where the 'Thereminists' of the time were essentially accomplished imitators of the violin instrument. But of course, many others have realized its properties as a unique instrument and created idiomatic works. Similarly, networks providing musical experience vary in their structure, and so in their purpose as well. Distributed Music Rehearsal project by Konstantas, Orlarey, Gibbs and Carbonel (1997), searches for ways of bringing remote musicians and a conductor together for rehearsal. With the change in the medium of communication, the usual organizational system during a classical music rehearsal is altered, and this situation brought uncomfortable moments reported by the performers; such as the wearying effect of having to constantly look into an illuminated screen, or reduced accuracy of the conductor when observed from the 2D display. Other researches on musical rehearsing and tutoring (Duffy and Healey, 2014) also mention the small flat screen as a negative impact on communication and performance due to losing spatial references, which led to verbal communication by the tutor as a resort. Unless new technological improvements are made in future, the loss of non-verbal cues and communication techniques most traditional musicians inherently use places network systems in an unfavorable position when it comes to teaching, practice and performance of traditional musics. This approach where the network is merely a 'Bridge' (Weinberg, 2003) is concerned with providing a technical aid to bring together performers. With this topology, the network scheme is not dealing with the improvement of any creative or collaborative processes, but this is not necessarily true in other types of networked music systems with different concerns.

3.3.1 Presence and intra-activity

Many network musicians seek to build live performance systems that contain the latency and communication drawbacks the above examples demonstrated. This inevitably leads to the issue of what 'live' music is, a matter of dispute in the field of electronic music ever since the genre first came to be. Since no physical contact is possible for musicians collaborating over networks, their presences need to be emphasized by other actions. For example, by setting up a topology so that musicians can interfere with each other on the net, as in an intra-active musical environment, a different kind of presence and togetherness can be realized. Implementations of this go all the way back to the League of Automatic Music Composers performances (Föllmer, 2005); where members would receive certain musical information from other players' sounds, impose changes on them, and send it forward. Another more recent example would be the Co-Audicle project, where multiple clients can work on the same piece of code in real-time.

One of the most important potentials of networked virtual instruments this project is concerned with is this concept of intra-activity. While all musical instruments (aside from purely generative systems) demand some sort of interactivity from the participant, interfering with another player's instrument is very atypical in traditional settings. Since there are no such physical restraints in virtual multimedia environments, the performers can intra-act much more easily by taking on each other's instruments and affecting the final output. Moore and Place (2001) point out the radical alterations intra-activity brings to ensembles; with this changed setting, the group of musicians will be considered as playing one inclusive instrument, which no individual has complete control over. The authors resemble this not as being part of an ensemble, but of an organism consisting of interdependencies among individuals, and building up to a singular audio-visual entity. Furthermore, with these intra-active designs, they signal the possibilities of new game-based models for musical performance leading to increased competition and new musical exploration. In addition to the element of unpredictability in both games and networked musics, the concept of intra-activity is also an overlapping factor in both practices.

Going back to the League example, the Internet was yet to appear at the time of the ensemble's performances; so all setups had to be local area network (LAN) connections, and all performers were situated in the same room, contributing in one way or another to the element of presence during the performance. So how must a network ensemble in the age of Internet and wireless transmission achieve this sensation? One performer, situated in front of an audience, staring down his/her laptop screen admittedly does not provide a sense of group collaboration to the audience. As discussed in the previous chapter, some musicians tend to incorporate different control mechanisms in their performance to increase the value of their tools as an instrument. However, in network performances, increased focus on personal gestures while playing as a band could shift the audience attention towards virtuosic actions of the individual instead of what the ensemble is doing collectively. It is a common practice among network bands to utilize a text-based communication system between each other, and some including the Hub and Glitch Lich have preferred to project the conversations for the audience to see in some of their performances. In addition, the visual presentation may not be limited with the chat log; for multimedia network music projects, displaying the graphical environment is also another option to bring the audience closer to the process, but this too does not always report an increase in musical appreciation from the audience (Hamilton, 2008).

3.4 Topologies of Network Musics

How the participants access and interact within these virtual spaces, the freedom they have, and their styles of expression strictly depend on the network topology, meaning the structure of the connection between the individual nodes. There are many ways to connect computer networks; some include connecting nodes in a ring (like some setups of the League concerts), connecting each client to a server to form a hub (similar to Monad application presented in this paper), or setting N-tiered networks with many hubs within (like the Internet itself).

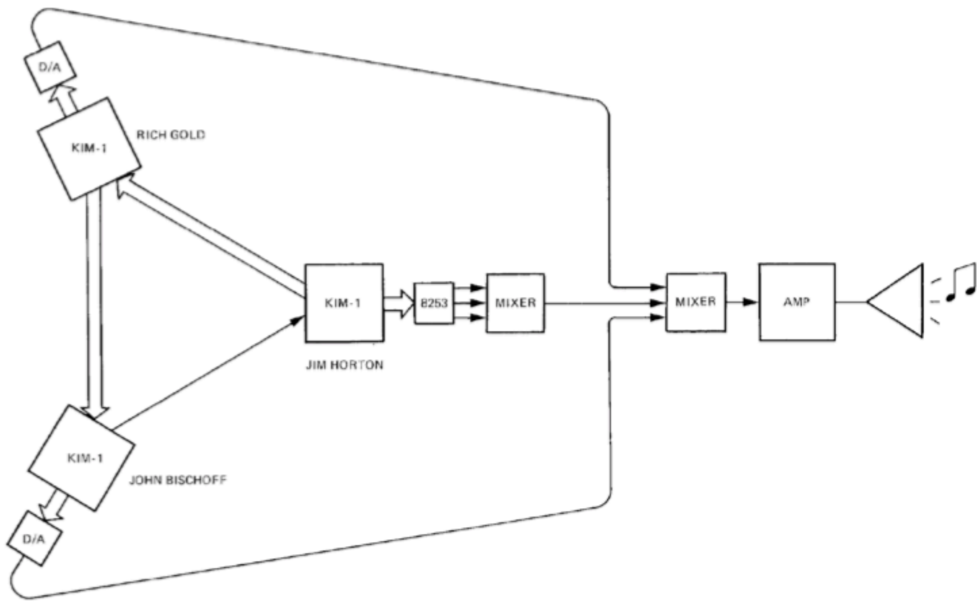

Figure 3.1 : The League's 1978 setup demonstrates a local ring network (Bischoff et al, 1978)

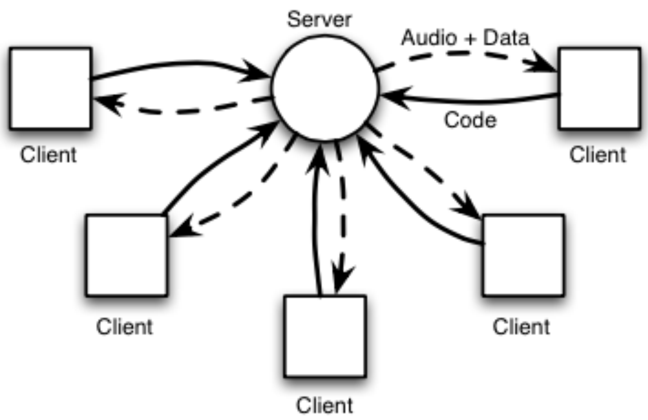

Figure 3.2 : Co-Audicle's client/server model (Wang et al, 2005)

Weinberg's (2003) approach towards analyzing network architectures contain much more social references when defining them. By separating what he calls Interconnected Musical Networks (IMNs) into two main groups, centralized and decentralized systems, the technical aspects are investigated hierarchically. In terms of group behavior, this classification suggests the social organization of the users. Since we can observe the presence or absence of a hierarchic structure in all collaborative music pieces, establishing IMN topologies analogous to social structures would at least contribute to 'humanizing' the engineering process. A centralized system for example, would take information from the players' input and send it to a center of activity, where the data will be analyzed and a musical result will follow. Decentralized systems on the other hand enable direct interaction between the participants determined by their respective computational capabilities. In addition to this grouping, the author points out another category to evaluate: the method in which nodes share information. Synchronous network topologies work towards real-time participation and manipulation, whereas sequential systems rather prepare events and submit to others after waiting for some time. Monad, to be mentioned more comprehensively in the coming chapters, demonstrates both synchronous and sequential operation under a decentralized system.

On the other hand, Rohrhuber (2007) argues the topology alone will not be enough to reflect the network music piece itself as a whole. Focusing on the logical organization of the network music piece may lead to different conclusions than of the actual performed product. Since systems with well-organized structure like telematic music pieces generally demonstrate an open and undetermined form during operation, the physical and acoustic space it is placed in is just as important when evaluating the device as a whole. All these aspects come together to create a 'causal topology' with inherent variations. This causation is the underlying factor in idiomatic networked music performances, providing interest and variety in compositions that can potentially surprise the audience and the performers, as well as the composer. The element of uncertainty regarding the individual perception of the work, in addition to the uncertainty stemming from network topologies gives a different personality to the music produced on network structures (Rohrhuber and de Campo, 2004); whether something happens due to intention, consequence or randomness is largely undetermined and open to interpretation. Unless the composer interferes with this situation to reduce the progress into a largely structured manner, improvised or quasi-improvised network bands take advantage of this aspect of topologies regularly.

It is this uncertainty present in both video games and network music performances this paper seeks to unify in order to create musical instruments or composition suitable for online collaboration. By utilizing computer graphics as a visual interaction tool similar to games and implementing rules and mechanics to play, telematic music compositions can contain elements of uncertainty, intrigue, challenge and even competitiveness within their communication-based structure. The paper will continue with some past and recent examples related to this approach, and demonstrate how online gaming can influence networked music compositions.

3.5 Past Network Music Projects

3.5.1 The League of Automatic Music Composers

MOS Technology launched KIM-1 microcomputers in 1976, and it immediately drew the attention of a group of artists coming from varying backgrounds living in the San Francisco Bay area. Among them, Tim Perkis, Jon Bischoff and Jim Horton formed "The League", by placing three KIM-1's in a triangular setting (no doubt an emphasis on non-hierarchy via equidistance), synthesizing sound, and sending each other sound information such as pitch or harmonic content. The process displayed a highly democratic manner of music making; there were no superiors within the network, and all operations were decentralized. This was a band of computers making sound independent from each other while at the same time aware of their collaborators; conceptually similar to any other musical band, yet unique in terms of the communication between.

The League of Automatic Music Composers 'manifesto', calligraphed by Don Day circa 1981, opens with the following quote:

> All that is not information, not redundancy, not form and not restraint – is noise, the only possible source of new patterns.
>
> Gregory Bateson

As the manifesto unfolds, one can notice a not-so-hidden ideological thought as well as technical and artistic motivation brought together within the League. In fact, members have often described the type of organization during their performances (given as informal concerts) as democratic and anarchic. Going back to Weinberg's definitions for IMN topologies, the 'anarchy' in performances of the League can be described as a result of its decentralized topology. Due to assigning different rules and parameters for each user to play with during the performance, the setup displays an unequal approach within a decentralized system, resulting in anarchy. The sound output also suits this structure, since it is highly disorganized, and cacophonic.

The commercial availability of KIM is without doubt the reason behind the incarnation of such an idea. Although the work in mention is the first of its kind, the bidirectional flow of influence between new -and affordable- technology and creativity has been the case throughout the history of electronic music. Continuing the League's initiative, many other computer network bands would form and compose with the technological means of their time.

3.5.2 maps and legends

Robert Hamilton's *maps and legends* is an appropriate example of network music composition and performance inspired by computer gaming. The work is centered on the use of a modified Quake III engine named q3apd to construct a virtual space where users can navigate in first person shooter (FPS) style. The actions in the game-like environment are then translated into sound and diffused in a physical space from a multi-channel speaker system. As the work's name suggests, the composer uses extensive mapping techniques to enable live data from the virtual environment to pass on to any number of sound generation software using Open Sound Control protocol (OSC) over User Datagram Protocol (UDP). Hamilton points out that the composer should take notice of the dynamic changes in the game environment since the sounds generated in the software(s) should provide "a satisfying and logical structure to the musical product" (2008).

The system is built on a network client-game server topology similar to the original Quake III game. Each player controls his/her own avatar's positions and actions, which is a proxy for producing the musical control parameters. The game server collecting this information streams the OSC data to a sound server, an 8-channel Puredata (Pd) patch, which in turn diffuses the sound to a custom built performance space. Since all performers/players interact through a visual interface, displays were presented to the audience in different ways like the viewpoint of one user, or a global vantage point. This approach challenged the listeners and caused confusion between the relationship of virtual and physical environments. Hamilton also notes the considerable change in audience reception when no visuals are displayed to them. In these performances, the audience demonstrated more focused listening towards the sonic output, and spatial perception was reportedly less muddled. But of course, the actions of the players within game environments are not meant to be concealed. On the contrary, their quasi-improvised play results in sonic causalities and are thus vital to the composition.

3.5.3 Shoggoth

Shoggoth is another network music program that employs 3D graphical environments and game-like controls for users. Instead of modifying an existing game engine however, Shoggoth's engine is built from the ground-up, taking much more time to fully develop (McKinney and Collins, 2013). But as a result, the system is customized specifically for network music performance situations. Within the virtual environment, the users generate different forms of meshes based on various algorithmic models such as cellular automata or flocking, and utilize wave terrain synthesis (Roads, 1996) on the modified mesh surfaces. The synths for sound output are designed outside of the environment using SuperCollider, forming a cross-software dependency. Avoiding peer to peer networking due to packet losses (McKinney and McKinney, 2012), Shoggoth uses OSC messages to communicate between clients and server, similar to maps and legends but utilizes a custom engine instead of modifying an existing game engine. This custom engine, called OSCthulhu, is further discussed in chapter 4.

3.5.4 Co-Audicle

Co-Audicle concentrates on musical cooperation of people in different locations towards constructing and playing one musical instrument. Co-Audicle's operation is based on the act of coding, which is stated as a type of performance and an expressive tool (Wang et al., 2005). The program is centered on ChucK, a time-oriented language for audio programming, but also complimented by CHUI framework for dynamic user interface design and GLucK toolkit, providing openGL, GLU and GLUT functionality for visual programming.

What is most interesting about Co-Audicle is that it can operate under different topologies: client-server and peer-to-peer. The client-server model assigns the initiating user as a server, making him/her a super user capable of setting rules and modes. The audio is also synthesized within the server and distributed to clients, while the clients, behaving as dumb terminals, feed the server with their individual or collaborative codes. The peer-to-peer model makes each node capable of synthesizing audio and running the Chuck virtual machine rather than just the server in the previous model. The convenience of peer-to-peer in Co-Audicle is that no audio is transferred between the nodes, only the code and meta-data. But, this brings other challenges to the collaboration such as timing between nodes, consistency across nodes, and the security of each node.

4. GAMES, NETWORKS AND FRAMEWORKS

4.1 How Games Work

All games contain specific structures and are confined in certain logic; be it a story, or a challenge. Forming logical rules and structure might be easy, but what makes a game appealing to a player? First of all, when someone desires to play a game, then he/she is taking on some kind of a challenge. Of course, these challenges bring along the motivation to overcome them. What sets them apart from other 'fixed' forms of entertainment is the element of unpredictability up to a certain degree, or simply the interactivity (Rouse, 2004). Games tend to provide different experiences each time they are played, even if the challenge is the same. This provides a unique kind of having fun, not present in any other forms of entertainment such as books or film. Within the rules it is built upon and the system it functions (i.e. gameplay), the user discovers and plays a game, leading to the ultimate goal of 'having fun'. But when designing games, there is a need for better, more specific description and categorization of 'fun'. Hunicke, LeBlanc and Zubek (2004) offer some taxonomy of gaming experience for this particular issue:

Sensation – Games as sense-pleasure

Fantasy – Game as make-believe

Narrative – Game as drama

Challenge – Game as obstacle course

Fellowship – Game as social framework

Discovery – Game as uncharted territory

Expression – Game as self-discovery

Submission – Game as pastime

With these classifications (and possibly more) of 'fun', we can better categorize games, be it computer or not, in their behavior, and the goals they demand from the user. The common game 'Hide and go seek' for example, would include

Challenge, *Discovery* and *Expression* (as well as *Fellowship* in some versions), which represent the dynamics of the game and 'fun' which results from it.

Games also provide social activity and preserve a communal spirit; almost all non-computer games require some form of socialization. In a social sense, computer games can be grouped into two: single-player and multi-player. While most computer games are designed as single-player, multi-player video games go as far back as 1958's Tennis for Two, and first networked multi-player games appeared in US universities' shared terminals in the 1970s. But with the dawn of the Internet in the next decade, the popularity of multi-player computer games would exceed expectations.

To evaluate the roles of both the game designer and the player, we can further consider the Mechanics, Dynamics and Aesthetics (MDA) framework presented by the aforementioned authors for analyzing games. Mechanics construct the modules of a game that form the decision-making structure (dynamic) to play in. Dynamics then determine how these mechanics are applied during gameplay. Staying with our hide and seek example, the mechanics of the game consist of the countdown (the time window to hide), game terrain (places to hide), and the search (seek), out of which dynamics like hiding, chasing and (in some versions) calling others emerge. These dynamic actions affect the senses and the total experience of the players, forming the aesthetics of the game, which are suspense, stealth, finding clues and shock (being found). So the game mechanics directly shape the dynamics, which bring out a certain aesthetic result to form an enjoyable and complete game.

4.2 Interaction in Virtual Environments

Virtual environments, synthetic worlds, cyberspace; however we wish to address them, they are software-generated spaces where about 3 billion, near half the Earth's population, interact with (Internet World Stats, 2014). These interactions bridge remote locations together in high speed, but often indirectly. A virtual environment presents us certain objects to perform an interaction with, which may result in connecting with other people. The 'things' that exist in these environments may or may not refer to real-world entities. At times they are merely representations of physical items like food or drinks that cannot function in cyberspace, in other cases there are entities that have no equals in the real world at all, like the mouse cursor, drop-down menus or scroll bars. Even though they are not physical, sometimes the experiences from these virtual entities provide real-world results, like playing chess, reading a book, or spending money (Brey, 2003). The pieces in virtual chess for example, cannot be touched, lost or damaged yet the rules, system and the fundamental game experience are preserved. The social communication during the game is altered; the player is unable to make eye contact with the opponent, or even identify who he/she is playing against: a person or a computer. In order to understand one another on the net, we need to utilize methods different than what the physical world offers. Efficient means of communication is crucial for both network gamers and network musicians to perform together.

Social scientists observe human behavior in various spaces extensively, and their broad findings are outside the scope of this paper. Nevertheless, we should address certain differences between the virtual and real spaces a network musician should be concerned with. The 'presence' of participants in these situations is in two places simultaneously. On the net, whether gaming or collaborating in musical activity, we look through different screens placed in remote locations, yet all participants are able to witness and experience a common sensation in the computer-generated environment. This non-localized common ground, cyberspace, compels us to readjust our presence. Thus, when the real and virtual spaces overlap, a smooth shift between appropriate communication modes is necessary for both the online gamer and the network musician to perform efficiently.

Participants in any kind of collaborative activity display certain types of behavior and utilize (or even invent) one or more methods of communication, with or without intention. Keating and Sunakawa's (2010) observation of 'participation cues' within a LAN party of online gamers noted the players' swift shifts between the real and virtual environments while communicating, interacting and organizing within the game's context. In a complex environment realized by code, the participants from the 'real world' tend to look for spatial indices to provide an understanding of the space. As similar to the physical world as the virtual space might intend to be, the interaction between players is different than of face-to-face communication. The facial gestures and cues are no longer effective, and the conversation rate and sharing information alters dramatically. The chat system, for example, enables the users to exchange far more articulate and well-defined information with the use of language and transferring thoughts with typing, but taking more time and effort than simple eye contact or a headshake.

4.3 Analyzing Digital Games

Popular online games set in virtual environments share common features that can be grouped methodically for a comprehensive analysis. Digital games are being increasingly noted as an area of study for researchers from a variety of disciplines (Brey, 2003; Carroli, 1997; Consalvo and Dutton, 2006; Cai et al., 2006). In turn, a guideline for avatar-based games set in imaginary worlds has become necessary, in order to observe not just the hardware or rule-specific aspects of the game, but also user experience and socio-cultural implications. Consalvo et al. (2006) offer a methodological perspective consisting of four categories to evaluate digital games in a wider scale: object inventory, interface study, interaction map and gameplay log.

In a designed software environment, virtual entities such as objects are presented to the player for interaction and exploration. Whether they can be interacted in multiple ways, multiple users, serve any purpose or purposes, and what they cost are subject to analysis in each game. As three-dimensional creatures, how we interact with the virtual world of computer games on a two-dimensional screen depend largely on the interface they are presented to us. Often while creating these digital games, the more complex and in-depth they are built, the more attention and praise they receive. Yet the same cannot be said for their user interfaces (UIs); the more complicated and crowded they are, the more frustrated players tend to get due to the negative effect on the gameplay. The UI gives us an idea of what information the designers consider critical: the level of freedom a player is given to explore, and what ‘path’ the player is moving on as the game advances.

A more dynamic and changeable feature of games is their interaction map. Depending on the game, the player’s overall experience and the story they develop can vary greatly each time they participate, due to the formulated interaction map of the game. Furthermore, some aspects of the interaction maps might even be left undiscovered by the majority of users. Analyzing the interaction map will indicate the borders of interaction and how (or if) they change in time. Additionally, in a sociological aspect, these maps reveal the cultural implications of the game such as stereotypes, gender roles or ideological marks.

The examination of the overall game world, or gameplay log, further broadens the analysis, where movements of avatars, the context of their conversations, what references they make, how the game saving system affects the story. Gameplay log indicates the variety of the virtual environment, where in complex worlds give way to interesting and open-ended situations. Music is also a crucial element in the gameplay logs. Grand Theft Auto series for example, offer a variety of radio channels containing many songs from many different genres, where two different players can experience the game while listening to completely different music. Thus, each person's total experience varies with his or her musical preferences, and their attitudes of engaging in the virtual environment differ as well (Miller, 2007).

4.4 Deriving Music from Games

Composing with game rules is by no means a new approach to music. Indigenous peoples such as the Inuit perform traditional vocal games mainly for self-amusement and not necessarily for singing, yet they still provide interesting musical results (Nattiez, 1989). During the 20th century, the experimental music composer Iannis Xenakis gave special attention to stochastic (meaning unpredictable) processes in his compositional models, and frequently employed game theory within his highly formalized music style (Arsenault, 2002). On the other hand, some composers of the time were getting involved in designing sounds and music for commercial games, like Suzanne Ciani's work for the Xenon pinball game, which included vocal expressions, female speech, synthesizers and many other sounds stored in computer chips attached to the game (crystalsculpture2, 2007). While music is not in the foreground or a primary concern for the pinball players, the process nevertheless contributes to an overall immersion of the participant inside the game, and at the same time produces non-linear sonic results. Each time the game is played, the sounds would combine in original ways, since every game unfolds differently. As for digital musicians and multimedia artists, composing *with* games and composing *for* games are not strictly unrelated areas. Sophisticated gaming consoles, in particular, offer powerful audio engines for non-linear audio events and sound effects. As they are meant to function in virtual environments, their technical foundation could also be of service to sound installations, networked music spaces and other new digital instruments (Schütze, 2003).

Music and sound in general play a vital role in games set in virtual environments, especially when they are open-ended and players are able to create their own stories during gameplay. The music composed for the game Spore, credited to Brian Eno, creates music generatively by forming rules determined by the players' decisions in the game, and not repeating itself but slightly vary every time. Similarly, Thief employs a generative audio system to ensure players do not listen to the same composition over and over but are offered a more realistic and non-linear sound world. But outside of gaming purposes, video can also accompany sound in order to augment an idea or communicate a story, where its total representation belongs to the composer's interpretation and mapping. Rudi's (2005) computer music video compositions, for example, searches for ways to bring together coherent musical and visual experiences, where video refers and at times clarifies the sonic events. His computer music game Construction Drive, sets up a virtual landscape in which players drive around the terrain, triggering sound events interactively. Composing by encouraging attitudes related to gaming possess certain difficulties the composer/programmer must be aware of. Since sound events are non-linear and unpredictable, a certain gameplay might produce undesirable musical results, putting off the user to ever try again. In cases like this the composer can secure a minimum quality of music that is "good enough", or point the player towards an ultimate goal: a motivation to play again.

4.5 Game Networking Architectures

In his notes regarding the architecture of the Unreal Engine, which he worked on for three years until the game's release in 1998, Tim Sweeney examines the client-server and peer-to-peer architecture models for games (1999). For example, maps and legends, described in the previous chapter, was based on the Quake engine which Sweeney points out as a client-server architecture; whereas another popular game of the time, Doom, employs a peer-to-peer model. He goes on to introduce his Generalized Client-Server Model (GCSM), which reduces the data exchange amount between devices and localizes more data in each client as a subset of the server state. To create network music engines where all users interact in a shared world, McKinney et al. (2012). offers a GCSM approach corresponding to music making. OSCthulhu, developed by the authors, is an open source software using data synchronization systems similar to games on an OSC-based platform. This software is mainly concerned with the speed of synchronization akin to game servers, “as the average network music server will deal with significantly less traffic than a gaming server, and thus can afford to be faster at the expense of being less efficient” (2012, p. 311). The authors also mention a different approach to composition is also necessary with this model. Instead of considering the musicality as a series of events, it is advised they think about compositional decisions as series of objects (as in object-oriented programming).

Still, divergences due to packet losses and latencies are noted while using the engine, which re-reminds us the idiomatic properties of networks. Divergences in different nodes result in increasingly differing musics rather than a synchronized whole. Monad, for example, avoids packet losses by sending only the initial state and live changes for drawing and synthesis over TCP. Nevertheless, for network musicians who wish to have a more fluid network communication over UDP, game state-based architectures such as OSCthulhu will also serve as a useful model.

5. MONAD: AN INTRA-ACTIVE NETWORK MUSIC ENVIRONMENT

Supporting the research and findings in the previous sections, a software was designed and developed for network music performances, named Monad (Çakmak, 2015). The software's primary aim is to bridge collaborative music-making experience with common video game rules. Monad is meant for musical performances in a virtual environment, with remote players joining from any part of the globe with a decent Internet connection.

Some initial objectives for the program prior to the design phase were:

- Create a 3D virtual environment where all players can navigate and explore.
- The players should have the ability to construct (or perhaps destruct) objects and/or structures in relation to music making.
- The players should have the ability to change and manipulate the objects and structures other players have made.
- The environment should be democratic. All players will have the same parameters to change and play with. No superusers.
- Include generative properties in the program.
- Musical results must be enjoyable or interesting, and convince the participants to play more.
- Synthesize the sounds during operation, rather than using pre-arranged sound samples.
- Musical output should not be concerned with harmonic relationship, although players can choose to push the composition toward a harmonic direction.
- Non-musicians or people without traditional music education should be able to play and enjoy as well.

- Include a text-based communication tool within the program (i.e. chat system).
- During the performance, certain game rules and mechanics should exist, and the players should be aware of it.
- Avoid 'survival' type gameplay (i.e. killing off players, trying to remain alive). The participants' primary concern must be to create music in a communal attitude.
- The program should not steer the players towards a pre-defined musical form or composition, the environment should be open and musically unpredictable.
- Network properties like latency and/or quality loss should not affect the performance or gameplay negatively, but rather increase the musical variety and encourage unpredictability.
- The visual and sonic material exhibited by the program should interest a passive audience and convince them to pay attention throughout the performance.
- The program should be capable of operating within browsers or via other common forms of Internet access.

With these ambitions in mind, I began researching ways to bring together some of these concepts and ideas together. It was evident that the program needed to be simple in its operation and display, yet engaging and logically well constructed.

5. 1 Tools Used

One of the most important conditions to build a game-like network music space was to give the connected player a sense of 'play' within the virtual environment. From relatively simple 2D platform games such as Super Mario to more sandbox-type nonlinear games like Grand Theft Auto, there is a sense of exploration while a player navigates around the computer game. Since Monad's environment was meant to be three dimensional, it was best that the participants explore the space in a first-person shooter (FPS) style camera movement, sans the weaponry. openFrameworks, an open source C++ toolkit, provides the OpenGL

Application Programming Interface (API), a vector graphics renderer frequently used by video game developers as well as offering interface design practicalities for computer music developers (Freed, 1995). The class ofEasyCam within the openFrameworks library enabled simple and customizable navigation in the openGL rendered 3D environment with the use of a mouse or track pad. For the user interface (UI) of the program I have used ofxUI, an openFrameworks addon developed by Reza Ali, which offers a wide range of sliders, buttons and other widgets such as text editor and text field for the chat window of the program. I have found this library a bit more extended and flexible compared to ofxGUI, the native library in openFrameworks. Monad's UI is designed to offer control over all parameters of the virtual structure as well as network communication with the chat window.

5.2 Visual Design

Since the program is not limited to pure sound but it is targeted as an audiovisual experience, the visual display and elements had to be carefully designed. These 'objects' or 'structures' would cause sound to emerge, which led my search to a closely related domain; graphical sound. Graphical sound's roots go back to a group of Soviet Russian inventors, scientists and musicians in the 1920s. The sound-on-film technology, synthesizing drawn patterns on a piece of film to sound by light, enabled sound to be captured in a plastic medium, further manipulated and changed. This breakthrough immediately drew the attention of the illustrators, composers and inventors working in the laboratories for sound movies. Namely Arseny Avraamov (chief of composer's brigade) and Evgeny Sholpo (inventor) initiated practical work towards drawing and synthesizing sound based on mathematical and acoustic data. It should also be mentioned that these workspaces were the precedents of computer music studios to arrive later in the century (Smirnov, 2013). As for graphical sound, many new instruments and devices would be invented as a result, in addition to new names and terminology to suit this new field. Evegeny Sholpo's Variaphone instrument series was one of these inventions; first patented in 1930 and four versions were built and continuously improved upon throughout the next two decades until the closing of the studio in 1950. Sholpo utilized optical rotating discs to generate sound and timbre by means of additive synthesis. The variaphone instrument was also the predecessor of the more widely known ANS synthesizer, developed by Evgeny Murzin, Oramics technique by Daphne Oram, and later in the century Iannis Xenakis' GENDYN device (Wilkins, 2013). A sketch for one of the experimental multi-track discs designed for the fourth (and unfinished) version of the Variaphone inspired the visual design of Monad. Moreover, the sonic behavior of optical discs also influenced the synthesis design.

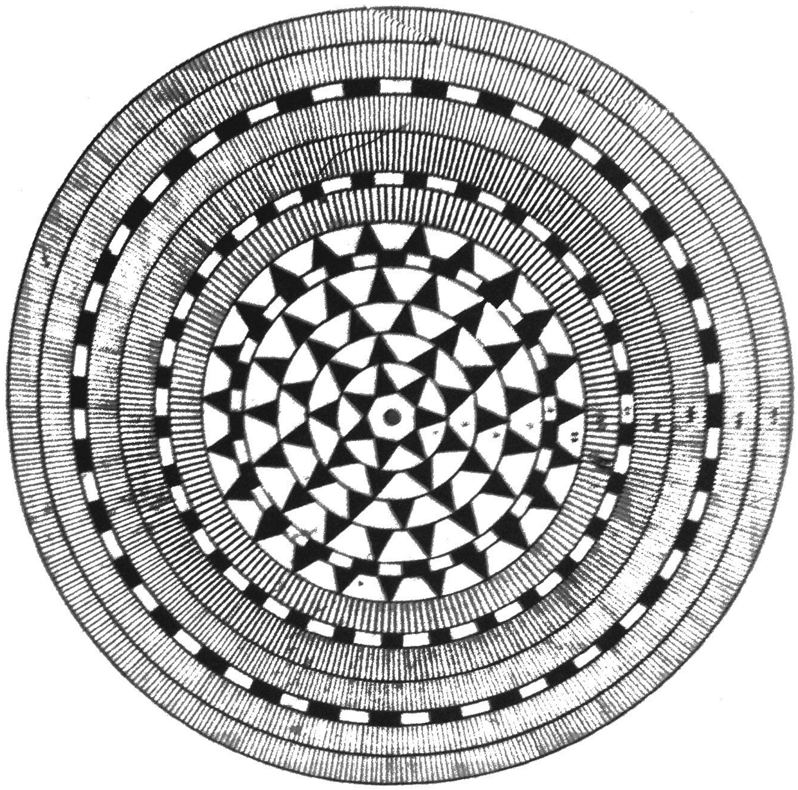

Figure 5.1 : An optical disc designed for the fourth version of Variaphone (Smirnov, 2013).

5.3 Graphics as Sound Objects

One of the advantages of virtual spaces is that three-dimensional physical laws of our nature need not apply, although they are frequently used for our cognitive and spatial referencing. Thus, the optical disc objects in real world can be transformed into sound objects, where each 'groove' layer can act more independent from the rest. In the acousmatic music tradition conceived by Pierre Schaeffer, sound objects are defined mainly as sounds which the listener cannot recognize its original source. Extending this concept later on (Roads, 2001), the term sound object referred to all sounds within a temporal limit. In this project, the sound objects are represented as virtual and visual objects created specifically to produce sound as they are manipulated. A rotating disc, or any other rotating object for that matter, immediately hints to a repetitive pattern or behavior. As a consequence, rotating the grooves in a cycle has also shaped the musical style and the sonic behavior of the program. Just like the difference of light going through the optical discs determine the sound change, the rotation of these virtual grooves would initiate a pulse and a frequency change in the virtual space. This pulsing sound strategy also partially eliminated the latency issues that could potentially arise when one of the players might initiate a rotation, but another player on another continent will most probably hear it slightly phased compared to the initiator's sounds. This pulsing strategy forms a unity in all connected clients of the program; the same pulse will go on regardless of its relative beginning in each node. This is further discussed in the section about time delays in Monad.

5.4 Mechanics, Dynamics, Aesthetics

This graphical and musical system also required some conditions and restrictions for the users to 'play' in. As stated in the initial objectives, it was important that the rules to play did not contain extra-musical winning conditions, which would consequently drive the participants away from music and towards a less aesthetically concerned approach. So these game rules needed to be concerned with leading the players towards determining the general musical form and structure: how to begin, how to proceed, and how to conclude.

For the program's game mechanics, a resource-based system was utilized to make changes and interact with the system. By setting costs for every change and subtracting the costs from a 'budget', meaning the current amount of resources, a certain 'economy' will function in the virtual environment. The main restriction this economy imposes is preventing the players from changing too many parameters hastily and encouraging them to pay more attention when spending their resources. But, while this method stops the participants from overplaying, it might also give rise to a situation where players become passive and reluctant to take action. During the design phase of these mechanics, it was also considered to punish passive players by reducing their budget after a time of inactivity. But this method would lead to an effort of 'balancing' the amount of actions per a given time for each player, and therefore bring in a survival concern to the game. Since enforcing this rule would in turn remove the players' main attention from music making by interacting only for the sake of keeping resources, it was later withdrawn.

To balance the 'spending' behavior of the players while making music and also to realize the communal spirit of making something, mentioned in the beginning of the chapter as an objective of the program, a rewarding system has been designed and implemented. On each player's screen during the performance, he/she will receive the most recent updates of other players' actions in the form of UI buttons in the color of the player it is related with. If the player finds the action musically appropriate, he/she can click the update button related to that action and thus give a certain amount of resources to that specific player without sacrificing their own budget. Players can benefit from this rewarding mechanism by receiving more resources than they spend, thus increasing their total budget by spending. There is also the case of rewards not matching the amount spent, as in the cases of choosing costly moves and not getting enough rewards from players, where the rewarding system acts more like compensation. There might also be instances where the rewarding mechanism is used only for the sake of keeping each other in the game, which would indicate a sense of togetherness within the group of players or the determination to keep the musical performance going. Both these resolutions target the intended behavior from the players in the first place.

As for the musical mechanics and dynamics of the system, Monad uses a purely synthesized approach to music-making part due to its existence in a synthetic environment. In line with Gregory Bateson's quote on the League manifesto mentioned in chapter 3, initial Monad tests used white and pink noise and heavy filtering to produce sounds and timbres. Although this resulted in a rich spectral mixture, the noise sources demanded an elaborate control structure to balance volume levels and clipping while playing with the filter. Later in the development phase, sawtooth wave oscillators replaced noise sources due to their high number of natural overtones; these overtones could be hidden and revealed with the use of bandpass filters, controlled graphically by adjusting groove radii. Similar to optical discs, the rotation speed and texture density affect the frequency of the pulse, where texture density also widens or narrows the volume envelope width. All these changes cost resources and reduce the player's total budget unless others reward them. There is little chance all players will empty out their resources in a multiplayer performance and can't get back in the 'game' (since players without resources can still reward others), so the players can virtually keep the performance going endlessly. The resource levels indicate the quantity of total actions, a sense of progress during the performance (leading to musical form) and a communal spirit where players help each other out and ensure they are involved.

5.5 User Interface

Monad offers a dynamic UI while playing. Each independent groove displays its own information in its own UI when selected. Through the UI, the player can assign texture, change rotation speed, modify the texture density, initiate/pause/reset z-motion, mute the object, delete all object information, toggle the chat window visible, or navigate to another groove. The user can also handle groove navigation by two assigned keyboard buttons (for moving inwards and outwards). With this option, users can keep one hand on the keyboard to rapidly switch between grooves, and the other hand on the mouse to change UI parameters, corresponding to an Atari-like sensation. The simplest UI canvas appears when the user selects no groove. Here, the only musical option is to globally initiate/pause/reset z-motion for all grooves, a rather costly, but musically the most distinguishable option. The significance and affects of the UI in music systems such as Monad will be further discussed in the evaluation section.

5.6 Networking Structure

openFrameworks also comes with the ofxNetwork addon for handling network communications. Since Monad's topology is a server-client relationship (where all players are clients and the server keeps the 'current state'), ofxTCPClient and ofxTCPServer classes have been used to send and listen to information. When the client is built and running, the TCP client class' 'setup' method is called, followed by immediately sending a 'hello' message to the server, who's IP is registered with the setup call before. This in turn sends all the current information required to draw and make sound on the client's system.

Monad's network topology is a relatively simple server-client relationship, or a single hub, where each player is a unique client, containing information like color, position, life and of course, IP number. The server stores all the control values for all operations done in the clients' program, but does not send any visual or sonic content for efficiency purposes. The server control values are for sending the correct drawing, movement and sound synthesis parameters on each client's computer. In that sense, the server can be also referred to as the shared object (Rohrhuber, 2007), which is altered from state to state by TCP messages coming in from connected players. While the server updates itself with every change reported from clients, it also acts like a 'wall', meaning it bounces off the client's message to every other connected client without changing or repackaging the contents. In essence, all nodes (including the server) in the network split and update the same message within their own system. The server's most important role is demonstrated when a client makes a new connection. Upon receiving the "hello" message from the client, the player's IP number is matched with other players to determine a reconnection. After this, the client receives all up to date values and changes in the program to draw and synthesize, as well as its own color information (new colors are given to new players) and how much resources he/she has (maximum amount of resources are given to each new player).

Figure 5.2: In-development and final versions of Monad UI.

5.7 Time Delays in Monad

A telematic music piece like Monad is designed specifically to function in an idiomatic manner within a networked environment of unavoidable latencies. Some network music systems attempt to overcome the latency issue via holding a global delay window larger than the network's causal delay in order to synchronize all players. However, others (Caceres and Renaud, 2008) are designed to perform in asynchronous network environments. The composition/software in each client contains one or more pulsing sounds most probably phasing in and out of each other. In addition to this, each new pulse by a client is also initiated in rest of the players' systems, starting with a slight offset at each client in terms of absolute time. As a result, the same sound is synthesized with the same period of occurrence in each computer, but in relation to other pulses is placed differently than of other players, noticeably or not. By using time delays as a musical technique, each node in the network contains rhythmic and phase variations deviating from others.

Furthermore, the z-motion demonstrated by the independent grooves (when told to 'move') also causes variation in each client, since each of their displacement will be slightly offset in any given global moment. A one-dimensional perlin noise object (Memo Akten's ofxMSAPerlin) is assigned to each groove to achieve independent and natural motion, the values for depth are then sent to frequency modulation and global reverb parameters for synthesis. To make sure the object moves with the same continuous signal values in each player, all clients receive the same seed numbers for identical motion generated by the server during initialize. The differences in displacement are removed when one of the clients order 'stop', which reports the final counter value and coordinate position to the server and every other client. While this action brings all clients in the same state relative to the object, receiving clients will notice a slight jump in image and sound when the groove is stopped and placed in its current position, compared to the continuous motion driven by the noise object.

Monad also displays sequential behavior in some cases, especially during the player's interaction with the UI elements. While changing a rotation speed, size or texture density by clicking and dragging the sliders, the player will observe immediate change in sound and image, but without releasing the mouse button the server and other clients will not be notified. This enables each player to have a small private space during the performance to find a situation closest to what he/she intended, and only after that inform other players and the server the change. One other advantage of this is to avoid sending too many TCP messages to other participants at the draw rate of the client, which resulted in overloading other clients and considerably slowing down the program, even freezing during initial tests.

5.8 User Experience and Evaluation

5.8.1 Two player tests and survey

Since Monad was designed as a musical performance device, it has been tested with a number of new users coming mostly from musical backgrounds. The participants were in remote locations and were instructed through the Internet, using a commonly available telecommunications application, Skype, which uses Voice over IP (VoIP), videoconferencing and instant messaging services. The tests began with the introduction of the application, and a 5-minute tutorial on how to use the camera, UI, and keyboard controls, how the game mechanics work and how to communicate with other players. Then, a two-player performance was carried out using only the application and without Skype. The duration of these trial performances varied from 15 to 40 minutes, depending on the user's wish to keep going. After the performance, the link to a 15-question online survey was sent to the participants to fill out. The survey consisted of rating various statements on a Likert-type scale (from 'definitely true' to 'not at all true') focused on the learning curve of the software, musical and gaming backgrounds of users, the efficiency of the user interface and the effect of the game mechanics. The evaluation of Monad is based on these survey results and personal comments of users.

As stated before, all participants have taken part in some kind of musical collaboration in the past, and most of them were professional musicians with different backgrounds and interests. Still, the self-rating of their gaming experience is noticeably divided, where half the total group claim to be experienced gamers and the other half rarely play computer or console games. In spite of this fact, all participants declared getting to learn how to use the program as easy. I gave the same verbal instructions to each user, where first they were introduced to the camera controls and navigation in the 3D space, then how to manipulate the grooves from the UI and their musical consequences, moving on to the game mechanics and finally how the economy works; all participants went through the same learning phases.

After the tutorial and during the performance, all users proclaimed they felt getting better as they played.

The survey results regarding the practicality of the UI controls in general displayed unclear results, where most participants neither firmly approved nor disapproved the UI controls. The ones with game design experience focused more on the UI use during their comments, and musicians tended to evaluate more on the sonic results. Another significant comment from almost all users when beginning the program was about the UI text field on the opening screen. Every participant was more or less surprised when they realized they were not able to copy-paste the server IP to the text field when initializing Monad. This is of course a very common habit for computer users on all platforms, and not being able to make use of this slightly displeased all participants before even starting to play.

Participants who are music professionals were more inclined to assess the expressive control features of the program. Instrumental musicians for example, demanded a more demonstrative interface such as joysticks or MIDI controllers, whereas gamers looked for more keyboard controls rather than using the mouse (the separation of expressive musical instruments and hermeneutical digital devices is an issue discussed more in-depth in chapter 2). The survey question asking whether the participants felt in control gave mixed results, where some have felt fully in control and some could not figure out what their actions changed as the music got more layered and dense.

The musical output during the performances were sonically similar to each other due to the program's restrictions on synthesis, but different users explored different musical directions. For example, gamers began with much faster movements and reactions and their changes were more rapid and arbitrary, sometimes causing the program to crash by accident. But as the environment filled with more sounds, they eventually slowed down to better observe the state. The musicians on the other hand, were significantly more careful and without haste when starting out, but their actions were more frequent once they were adjusted to the environment. There were also stylistic differences among musicians, where some utilized the pulsing actions similar to 20^{th} century American minimalism, while others tried to imitate more danceable grooves and rhythms, and some even shortened and speeded up the pulses, working towards sounds akin to granular synthesis. Whatever style the participants chose to impose, the survey results show a common satisfaction from the musical output.

Similar to the variation of musical results from person to person, the reactions to the game-like rules were also different. Some participants, especially musicians did not prefer the spend-reward mechanism due to its distraction from complete musical focus. As a result they tended to use the rewarding mechanism much later in the game, where the other player was almost or already out of resources. Another interesting fact was that all players immediately and intuitively referred to the budget as 'life', and the act of losing life as 'dying'. This is most likely due to all participants' familiarity to the now popular gaming culture conventions.

The survey gives a clearer picture regarding the gaming sensations such as competition, challenge, surprise and fellowship. First of all, none of the participants felt competitive against other players and strongly disagreed with the statement, with an exception of one player (from a musical background) who felt strongly competitive. This implies Monad's music making practice encourages collaboration and togetherness rather than competition and rivalry. However, this did not mean users were not challenged; the majority of players admitted they felt a creative challenge during play. Both music performances and games feature some level of 'challenge', but in different aspects; Monad's challenge leans toward the musical side, a creative one, rather than a competitive challenge found in most games.

The rewarding mechanism implemented in Monad came out of a desire to bring together musical choices with gaming conventions that would shape the 'gameplay', the duration and the form of the performance. Contrary to what was expected, participants paid little attention to the contents of the rewarding mechanism that presented as labeled dynamic buttons (i.e. "Groove 7 rotation set to 4.3906"). Gamers more often tended to reward other players in order to keep the game going, and without focusing on their individual actions written on the buttons. One of the gamers for example, started to rapidly change textures on a groove, causing noticeable changes in timbre while emptying his resources. In order to keep him 'alive', I had to stop making musical decisions and serially reward him resources. Musically, this resulted in a kind of 'solo', due to his groove's changing sound becoming more noticeable, as well as the musical inactivity of the other player. Musicians' attitude towards the rewarding mechanism was based significantly more on aesthetic choices of other players. Although many admitted they did not read the labels on the reward buttons, some pointed out they followed the changes with their ears rather than reading it off the screen, and decided to give resources based on sonic results. Another evidence indicating success, in terms of gaming sensations, is that all participants were surprised by some of the events during the performance. The element of surprise is crucial in games, where the overall experience is often improved when players come across unexpected situations that catch them off guard.

Finally, there was a strongly positive and unanimous settlement among the participants when they were asked whether they would like to play it again. It is common among gamers to play the same game over from the beginning, experiencing a different story while in the same environment as before. Musicians also encounter similar situations when they perform a piece live; each take is a different experience, especially for improvisatory bands like jazz ensembles. Monad's design and implementation focused on collaborative music making with game-like rules in a virtual environment. It can be said that the initial objectives were largely accomplished and successful. Still, the UI can be improved and better applied to fit the playing attitudes of all players. Furthermore, the system can use another degree of freedom for more advanced players to preserve its element of 'fun', and avoid the possibility of getting dull after a few more plays.

5.8.2 Multiplayer performance

Monad was performed with 4 players joining from Istanbul, Berlin and Vancouver with the server and audience located in Istanbul on May 24th 2015. Two days before the performance, the participants gathered for a 'dress rehearsal', and attempted to perform with this many players for the first time. The difference between hosting four clients rather than two were quickly noticed. First of all, the chat system referred to each person with his or her own IP address. While the player running the server was quickly identified (due to IP number 127.0.0.1), the others were not discernable. This issue was resolved with a quick adjustment so that all players were able to decide their own nicknames before entering the environment, and these nicknames represented them in the chat window.

Another difference of having many players at the same time was that the rewarding system did not function as efficiently as it did with two player trials. This is part due to the increased amount of activity in any given time; players were having a hard time concentrating on their own musical activity while trying to keep up with the other three participants. This irritated the group in general, especially when one player lost track of her resources and emptied it without realizing, and was left out of the play. Another player, coming from a gaming and programming background, tweaked his version of the code in order to never lose resources; in game terms, he basically cheated and switched to a permanent 'god mode'. Meanwhile, the others became distracted and consistently kept rewarding each other points in order to stay 'alive', and their musical actions became a lesser priority. Still, the rehearsal stretched out over 70 minutes and all players seemed highly motivated to keep going.

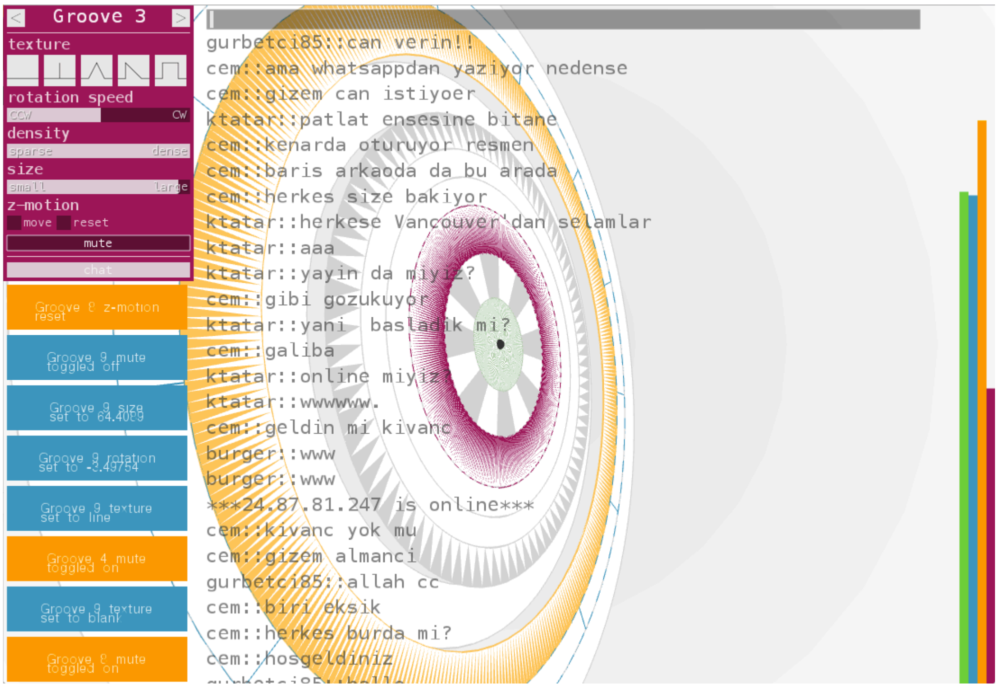

Figure 5.3: Monad screen view during performance

As stated before, the total amount of resources were intended to determine the overall duration and form of the piece. But in the concert setting, the maximum allowed length of performance was 15 minutes, so the game mechanics did not have the opportunity to dictate the performance. Thus, the resource-based economy seemed non-relevant to the musical progress. In order to implement the gameplay as an integral part of a performance with limited duration, the game mechanics must be further developed and tested.

The audience reaction to the performance was predominantly positive. The concertgoers were especially entertained by observing the chat window, which visualized banter-like dialogues between musicians. Since the viewers were not used to witnessing text-based communication among live performers, this drew in their attention significantly. Moreover, big screen projection of the graphical environment from the point of view of one player proved to be useful in terms of achieving presence, and the liveness of the performers. But, the audience later commented that after a certain amount of progress, they could not figure out what the actions of the players caused musically and they got lost. As mentioned in the previous section, some of the participants in the two player tests also had similar comments about losing track of individual actions as the activity in the environment increased. The performance kept all the viewers' interest nonetheless; the major facts were the chat log, a graphical/musical environment the viewers could follow, and the fact that remote performers were included.

6. CONCLUSION AND FUTURE WORK

Forming networked relationships in music and gaming lead to unique experiences. In both activities there are certain types of communication and collaboration that seem to overlap. Interactivity (or intra-activity) and the element of uncertainty are present in both practices, sometimes due to player interactions and at other times due to network effects. Developing new digital instruments for the purposes of network performances require an understanding of the proposed dimension spaces for defining, grouping and evaluating new musical devices. Furthermore, taking note of online gaming customs and their technical structures can offer new methods when designing the rules and dynamics for new collaborative virtual instruments. Growing into creative tools that contain more references towards multimedia practices rather than past musical traditions, digital instruments can remove themselves further from affiliation with conventional acoustic ones, and gain their own value as instruments.

Supporting the research on building networked music performance systems and bringing in the aforementioned ideas and concepts, a networked music environment named Monad is presented. Monad implements FPS type navigation, resource-based actions, text communication and rewarding mechanisms inspired by games to create a collaborative and intra-active virtual platform for music making. After testing the program, some suggestions for network instrument designers can be made:

- There needs to be a careful balance between characteristic limitations and game-like discoverability. Even if there are rules similar to games, one should avoid imposing too many of them on players. Diversity of expression should be encouraged.
- Players should not be directed towards an ultimately non-musical goal. While this could immerse the participants further in the virtual environment, the

generated sounds will become causal and removed from musical intentions.

- The program should present a clear and friendly user interface. If the users cannot get past a confusing group of sliders, buttons and labels they will not be able to fully understand the environment's dynamics.
- It is crucial that the players have some sense of agency in the program. They should be able to immediately observe the consequences of their actions, rather than trying to figure out what part is their making.

Internet use and related conventions are internalized by the more recent generations. Thus, live musical collaboration utilizing the Internet as a creative medium is of interest to both active participants and the new generation of electronic music/media audiences. After testing and preforming the software built to compliment the matters of this paper, it is evident that both the musicians and the audience take interest in new, collaborative, and networked music software. It is recommended for future work conducted in this area to focus primarily on bringing music-making and gaming activities closer together, where performers will not be able to strictly separate one from another while using networked programs. Some suggestions would be to challenge users during performance, without targeting pre-determined compositions. The players can be given time windows to perform a given amount of activity, within which they can be challenged as well as encouraged to be expressive and creative. Furthermore, similar to games, performers can play with (or against) artificial intelligence; since these environments will often be limited and exert their own characteristics, learning and performing mechanisms would be easier to implement within these musical environments compared to traditional instruments.

REFERENCES

Arsenault, L. M. (2002). Iannis Xenakis's "Achorripsis": The Matrix Game. *Computer Music Journal*, 26(1), 58-72.

Barbosa, A. (2003). Displaced Soundscapes: A Survey of Network Systems for Music and Sonic Art Creation. *Leonardo Music Journal*, 13, 53–59.

Barbosa, A. (2006). Computer-Supported Cooperative Work for Music Applications, *PhD thesis*, Music Technology Group Pompeu Fabra University, Spain, Barcelona.

Bengler, B. and Bryan-Kinns, N. (2013). Designing collaborative musical experiences for broad audiences. In *Creativity & Cognition*, p. 234-242. Eds. Nagai, Y. and Ferguson, S.

Berio, L. and Dalmonte, R. (2007). Intervista Sulla Musica. Laterza, Bari, Italy.

Birnbaum, D., Fiebrink, R., Malloch, J., Wanderley, M. (2005). Towards a Dimension Space for Musical Artifacts. In *NIME 2005 Proceedings*.

Bischoff, J. (1991). Software as Sculpture: Creating Music from the Ground Up. *Leonardo Music Journal*, 1(1), 37-40.

Bischoff, J., Gold, R., Horton, J. (1978). Music for an interactive network of microcomputers. *Computer Music Journal*, 2(3), 24–9.

Bongers, B. (1998). An Interview with Sensorband. *Computer Music Journal,* 22(1), 13-24.

Brey, P. (2003). The Social Ontology of Virtual Environments. *American Journal of Economics and Sociology*, 62(1), 269-282.

Cáceres, J. and Renaud, A. (2008). Playing the network: the use of time delays as musical devices. In *Proceedings of International Computer Music Conference*, 244-250, Belfast, Northern Ireland.

Cai, Y. Y., Lu B. F., Fan Z. W., Chan C. W., Lim, K. T., Qi, L., Li, L. (2006). Proteins, Immersive Games and Music. *Leonardo*, 39(2), 134-137.

Cascone, K. (2004). The Aesthetics of Failure: 'Post-Digital' Tendencies in Contemporary Computer Music. *Computer Music Journal*, 24(4), 12-18.

Carroli, L. (1997). Virtual Encounters: Community or Collaboration on the Internet. *Leonardo*, 30(5), 359-363.

Chafe, C. (2009). Tapping into the Internet as an Acoustical/Musical Medium. *Contemporary Music Review*, 28(4), 413-420.

Çakmak, Ö. C. (2015). monad_client. GitHub repository, https://github.com/occak/monad_client

Çakmak, Ö. C. (2015). monad_server. GitHub repository, https://github.com/occak/monad_server

De Campo, A. and Rohrhuber, J. (2004). Waiting and Uncertainty in Computer Music Networks. In *Proceedings of the International Computer Music Conference*, Miami, FL.

Duffy, S. and Healey, P. (2014). The conversational organization of musical contributions. *Psychology of Music*, 42(6), 888-893.

Freed, A. (1995). Editor's Notes: Improving Graphical User Interfaces for Computer Music Applications. *Computer Music Journal*, 19(4), 4-5.

Follmer, G. (2005). Electronic, Aesthetic and Social Factors in Net music. *Organized Sound*, 10(3), 185–192.

Golub, A. (2010). Being in the World (Of Warcraft): Raiding, Realism, and Knowledge Production in a Massively Multiplayer Online Game. *Anthropological Quarterly*, 83(1), 17-45.

Gresham-Lancaster, S. (1998). The Aesthetics and History of the Hub: The Effects of Changing Technology on Network Computer Music. *Leonardo Music Journal*, 8, 39-44.

Hamilton, R. (2007). Maps and legends: FPS-based interfaces for composition and immersive performance. In *Proceedings of the International Computer Music Conference*, 344–347, Ljubljana, Slovenia.

Hugill, A. (2012). Networked performance. In *The Digital Musician*, p. 162-168. Routledge, New York, NY

Hunicke, R., LeBlanc, M., Zubek, R. (2004). MDA: A formal approach to game design and game research. In *Proceedings of the Challenges in Games AI Workshop, Nineteenth National Conference of Artificial Intelligence*, San Jose, CA.

Keating, E. and Sunokawa, C. (2010). Participation Cues: Coordinating Activity and Collaboration in Complex Online Gaming Worlds. *Language in Society*, 39, 331-356.

Konstantas, D., Orlarey, Y., Gibbs, S., Carbonel, O. (1997). Distributed Musical Rehearsal. In *Proceedings of the International Computer Music Conference*. Thessaloniki, Greece.

Makelberge, N. (2012). Designing Instruments Towards Network Music Practices, *PhD thesis*, Universidade Católica Portuguesa, Lisbon, Portugal.

Makelberge, N. (2012). Rethinking Collaboration in Networked Music. *Organised Sound*, 17(1), 28-35.

Magnusson, T. (2010). An Epistemic Dimension Space for Musical Devices. In *Proceedings of the 2010 conference on New interfaces for musical expression*, 43–46.

McKinney, C. and McKinney, C. (2012). OSCthulhu: Applying video game state based synchronization to network computer music. In *Proceedings of the International Computer Music Conference,* Ljubljana, Slovenia.

McKinney, C. and Collins, N. (2012). Liveness in Network Music Performance. Date retrieved: 6 January 2015, address: http://www.chadmckinneyaudio.com/WP-Content/resources/papers/Liveness2013.pdf

McKinney, C. and Collins, N. (2013). An Interactive 3D Network Music Space. In *NIME 2013 Proceedings*.

Miller, K. (2007). Jacking the Dial: Radio, Race, and Place in "Grand Theft Auto". *Ethnomusicology*, 51(3), 402-438.

Mizuno, M. (2011). On the Music through Network. In *Proceedings of Asia Computer Music Project*, Tokyo, Japan.

Moore, S. and Place, T. (2001).KromoZone: Intermedia and Improvisation for the 21st Century. In *Music Without Walls? Music Without Instruments? Conference*. DeMontfort University, Leicester, UK.

Oliveros, P., Weaver, S., Dresser, M., Pitcher, J., Braasch, J., Chafe, C. (2009). Telematic Music: Six Perspectives. *Leonardo Music Journal*, 19, 95 - 96.

Pressing, J. (1990). Cybernetic Issues in Interactive Performance Systems. *Computer Music Journal*, 14(1), 12-25.

Roads, C. (1996). The Computer Music Tutorial. MIT Press, Cambridge, MA.

Roads, C. (2001). Microsound. MIT Press, Cambridge, MA.

Rohrhuber, J. (2007). Network music, in *Cambridge Companion to Electronic Music*, p. 140–155, Eds. Collins, N. and d'Escrivan. J., Cambridge University Press, Cambridge.

Rouse, R. (2005). Game Design: Theory & Practice, 2nd ed. Wordware Publishing, Plano, TX.

Rudi, J. (2005). Computer Music Video: A Composer's Perspective. *Computer Music Journal*, 29(4), 36-44.

Rudi, J. (2015). Past and Current Tendencies in Technology-Based Music. *Organized Sound*, 20(1), 30-36.

Schütze, S. (2003). The creation of an audio environment as part of a computer game world: the design for Jurassic Park – Operation Genesis on the XBOX™ as a broad concept for surround installation creation. *Organised Sound*, 8(2), 171-180.

Smirnov, A. (2013). Sound in Z: Experiments in Sound and Electronic Music in Early 20th Century Russia. Walther Koenig, Cologne, Germany.

Tanaka, A. (2006). Interaction, experience and the future of music. In *Consuming Music Together*, B. O'Hara, K.; Brown, Ed., **35**, *Computer Supported Cooperative Work*. Springer Netherlands, 267–288.

Thorne, S., Black, R., Sykes, J. (2009). Second Language Use, Socialization, and Learning in Internet Interest Communities and Online Gaming. *Modern Language Journal*, 93, 802-821.

Wanderley, M. and Orio, N. (2002). Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI. *Computer Music Journal*, 26(3), 62-76.

Wang, G., Misra, A., Davidson, P. and Cook, P. (2005). Co-Audicle: A Collaborative Audio Programming Space. In *Proceedings of the International Computer Music Conference,* Barcelona, Spain.

Weinberg, G. (2003). Interconnected Musical Networks - Bringing Expression and Thoughtfulness to Collaborative Group Playing, *PhD thesis*, MIT, MA.

Wessel, D. and Wright, M. (2002). Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3), 11–22.

Wilkins, I. (2013). Enemy of Music. In *Matter of Contradiction Conference*, London, England.

Consalvo, M. and Dutton, N. (2006). Game analysis: Developing a methodological toolkit for the qualitative study of online games. *Game Studies*, 6(1). Date retrieved: 1 April 2015, address: http://www.gamestudies.org/0601/articles/consalvo_dutton

Sweeney, T. (1999). Unreal Networking Architecture. Date retrieved: 2 May 2015, address: http://udn.epicgames.com/Three/NetworkingOverview.html

crystalsculpture2. (2007). *Omni-Suzanne Ciani* (Video). Date retrieved: 17.05.2015, address: https://www.youtube.com/watch?v=yC8O5HPvfsg

Url-1 < *http://www.theesa.com/wp-content/uploads/2015/03/ESA-2014-Annual-Report.pdf*>, date retrieved 30.04.2015.

Url-2 < *http://www.mpaa.org/*>, date retrieved 25.04.2015.

Url-3 <*http://www.internetworldstats.com/stats.htm*>, date retrieved 29.04.2015.

Url-4 < *https://www.mobygames.com/game/windows/spore_/credits*>, date retrieved 16.05.2015.