Department of Music
Norwegian University of
Science & Technology

Department of Musicology
University of Oslo

A Human-Machine Music Performance System based on Autonomous Agents

Pedro Pablo Lucas Bravo

May, 2022

Master's programme in Music, Communication & Technology

Abstract

This thesis proposes the design and implementation of a multimodal system for human-machine music performances in real-time. The machine behavior is modeled under the concepts and paradigms related to an artificial *Swarm of Autonomous Agents*. The system used three advanced technologies as subsystems: *Motion Capture*, *Spatial Audio*, and *Mixed Reality*. These subsystems are integrated in one only solution that is evaluated regarding *system measurements* and *music improvisation sessions*. The *system measurements* determine the advantages and limitations in terms of effectiveness and efficiency; and the *music improvisation sessions* evaluate user interaction through the analysis of data recording and a survey. The results provide latency, jitter and other real-time parameters that are contrasted with user data. Moreover, the user analysis shows that the system is easy-to-use and highly enjoyable. These findings indicate that the strategy to conceive the system is validated and can be used for further investigation for autonomous agents and musicology aspects[1].

[1] The complementary material (video demo, sound recordings, and source code) can be found in this blog post from the MCT website

Acknowledgement

I would like to express my deepest gratitude to my supervisor, Stefano Fasciani, for his support of this work and his professional advice to approach this thesis in the right direction.

Special thanks to all the people from the Master's programme in Music, Communication and Technology (MCT), teachers and classmates who inspired me in various ways through this journey.

Lastly, I am thankful to my family for the remote emotional support while I was in a distant land away from my beloved Middle of the World.

Contents

Chapter 1

Introduction

This thesis presents the design, implementation and evaluation of a real-time music system in a multimodal setting capable of providing interaction with artificial entities for improvising music in a human-machine context.

The modalities that were considered are focused in spatial information of a *sound source* in terms of *aural*, *visual*, and *body* communication between the user (music performer) and the system. This *sound source* contains musical material that the user improvises for a period, then the user decides to finish and loop the source which can be moved in space physically in order to be heard and seen in different locations. This work refers to this *sound source* as a *musical agent*, since the user can decide to activate an autonomous behaviour in terms of motion and music performance, thus in this context, an agent is considered an artificial musician that is moving around the human performer and modifying the musical material that was first played.

In this system the performer deals with a swarm of agents (sound sources). Each agent is aware of the human performer and other artificial individuals so that the resulting musical piece is an spatialized multi-layer improvisation.

This manuscript starts in this first chapter describing the problem statement through the motivation and research question, as well as the objectives and contribution to the field of music technology. After establishing the context of this thesis, key concepts will be presented as a background for the development of this work along with the relevant related literature showing the current state-of-the-art. Chapter 3 explains the evaluation methodology in several stages, then Chapter 4 illustrates in detail the requirements, design, and development of the system. The results of the study following the proposed methodology are shown in Chapter 5. Finally, a critical discussion, conclusions, and future work are presented in Chapters 6 and 7 respectively.

1.1 Motivation

We experience the world as a combination of different types of information received by our senses constantly. In music it is not the exception, specially when it comes to perform a piece as a group of musicians where each one is responsible of a part that harmonizes with the overall composition. In that sense, every musician hear, observe, and move regarding what is happening in the performance. As humans, we perceive the location of the musicians in such performance in terms of sound and images, in which the arrangement of the position for every of them is even chosen in a way that justify the spatial composition of the audio environment regarding sound balance (Turner, 2009).

Giving the advancements in technologies and strategies involved in contemporary music and other creative fields, this work is motivated by the experience described above by bringing a novel approach for allowing a human-machine interaction in a multimodal setup, where a human musician can interact with other artificial musicians that participate in terms of music performance and movement.

Additionally, the way to develop and explore a technological platform that fulfills such experience motivates the author to dive into the connection between the physical and virtual world through real-time digital solutions. Although it can bring several questions regarding: *how such platform can be designed and implemented*, *if it would work as intended*, *to what demographic*, *if it can be enjoyable*, and specially *how the resulting musical material would be*; the main interest of this work lies in the system that can be develop under the idea presented so far, establishing a base line for future exploration once such system is proved to be feasible.

Having in consideration this context, the description of the research work developed in this thesis is presented below.

1.2 Research Question and Objectives

At the present date, there are several works regarding the design and implementation of strategies supported by technologies and algorithms for music interaction and generation, which are conceived under physical and/or virtual environments. These works are mentioned in section 2.6 as related literature relevant to this thesis.

A human-machine musical performance experience with spatial characteristics, as the one proposed in this thesis, has not been explore before — to the knowledge of the author. A platform that supports such experience demands the design and implementation of an *effective* and *efficient* system for real-time usage since its target is music improvisation, which means, composing and performing on the fly.

It brings the statement of the following research question:

How can we design and implement a system for human-machine live music performances in a multimodal environment?

This question can cover a range of possible designs and thus it is necessary to define the scope of this thesis. This scope is reflected in the requirements and decisions — which come from the literature and the author's expertise — that support the proposed system presented in Chapter 4. In that sense, the following research objectives needs to be fulfilled:

1. Develop a multimodal system for music improvisation which implies to:
    - Define requirements and scope of the design and implementation.
    - Design the solution based on literature and the author knowledge.
    - Explore advantages and limitations of *Motion Capture*, *Spatial Audio*, and *Mixed Reality (MR)* technologies.
    - Explore integration challenges for the technologies mentioned before.
    - Implement the solution in a suitable technological platform that combines the mitigated technologies
2. Evaluate the *effectiveness* and *efficiency* of the solution through objective measurements from the implemented system.
3. Identify user behaviours and evaluate experience through anonymized data recorded in music performance sessions, surveys, and reflections.
4. Determine the advantages and limitations of the solution upon the results, and provide recommendations for further research and development.

These objectives are focused on the development and validation of the proposed system in order to confirm the design and answer the research question. Moreover, it requires the understanding of the background found in Chapter 2 which supports this work, as well as other concepts that will be explained in the corresponding parts of this manuscript.

1.3 Contribution

In this work, a technological environment is developed and studied as a tool for music expressiveness, which is relevant to the fields of *Interactive Music Systems (IMS)*, *Multi-Agent Systems*, and *Music in Extended Realities (Musical XR).* It expands the possibilities to create and evaluate methods and mediums for musical actors. The contributions of this thesis are the following ones:

- The design and implementation of the proposed system through the integration of *Motion Capture*, *Spatial Audio*, and *Mixed Reality (MR)* technologies.
- The strategy to evaluate relevant aspects for similar solutions.
- The results of the study for validating the system.

Being established the overall perspective of this research work, the next chapter describes the background and related work as starting point for the understanding of this thesis.

Chapter 2

Background

This chapter presents fundamental concepts in terms of abstractions, strategies, and previous work relevant for the development and the study of a solution that involves human-machine interaction in the context proposed in this thesis.

2.1 Musical Agents

Tatar & Pasquier (2019) define a *Musical Agent* as "an artificial entity that partially or completely automates musical creative tasks". For explaining the concept of an agent, they take and adapt the definition from Computer Sciences, in which it is seen as "an autonomous system that involves actions and responses within an environment along the time".

In terms of musical agents, Murray-Rust et al. (2006) present an architecture that uses the theory of *Musical Acts* taken from the concepts behind the *Speech Act* linguistics. They mention that the "human-like" agent properties promote a dynamic environment for human interaction among participants (human and artificial) over diverse musical material. Their framework is composed of well-defined rules for interactivity based on *communicative acts* so that different forms of interaction take place among agents, which brings to the notion of *Multi-Agent Systems* as well as the application of *Swarm Intelligence* for creating music.

In that sense, a *Multi-Agent* approach complements the concept mentioned initially when two or more entities are involved in an environment. Wulfhorst et al. (2003) use this approach for interactive music systems in which *Perception*, *Cognition*, and *Musical Execution* are the main abilities that musical agents should address for interactivity. Moreover, Tatar & Pasquier (2019) presents a topology to frame musical agents from the lowest to he highest level of autonomy which includes: *reac-*

tivity, *proactivity*, *interactivity*, *adaptability*, *versatility*, and *volition & framing.* This topology is summarized from simple responses to more detailed explanations regarding agent's behavior.

Musical agents over multi-agent design can be reinforced with other important abstractions from *Live Algorithms* and *Swarm Intelligence* as illustrated in the following sections.

2.2 Live Algorithms and Improvisation

For some situations, the representations and behaviours for musical agents are given in a full-autonomous setting, nevertheless, when humans interact with artificial entities in a musical context it is necessary to structure solutions to allow coexistence. From this perspective, Blackwell et al. (2012) developed the concept of *Live Algorithm* which is defined as "an autonomous music system capable of human-compatible performance... the Live Algorithm listens, reflects, selects, imagines, and articulates its musical thoughts as sound in a continuous process".

This section is supported mostly by the work from Blackwell et al. (2012) in which the "performance" part of the previous definition refers to *musical improvisation.* Improvisation demands skills to recognize constraints and also generate musical material spontaneously (Hermelin et al., 1989). An improvisation session can be carried out on the fly and participants can look at their executions and responses for continuous creation.

This form of *collective free improvisation* is proposed by Blackwell et al. (2012) as an ideal context for *machine improvisation* because it can be seen as an exchange of sonic events between people and machines, both as data sources.

Considering that Live Algorithms work over a collective human-machine musical improvisation, there are four attributes that are part of them as similar to humans: *autonomy*, *novelty*, *participation* and *leadership.* All of them are enclosed in the *PQf architecture* for computer music systems proposed by Blackwell (2007) as depicted in Figure 2.1.

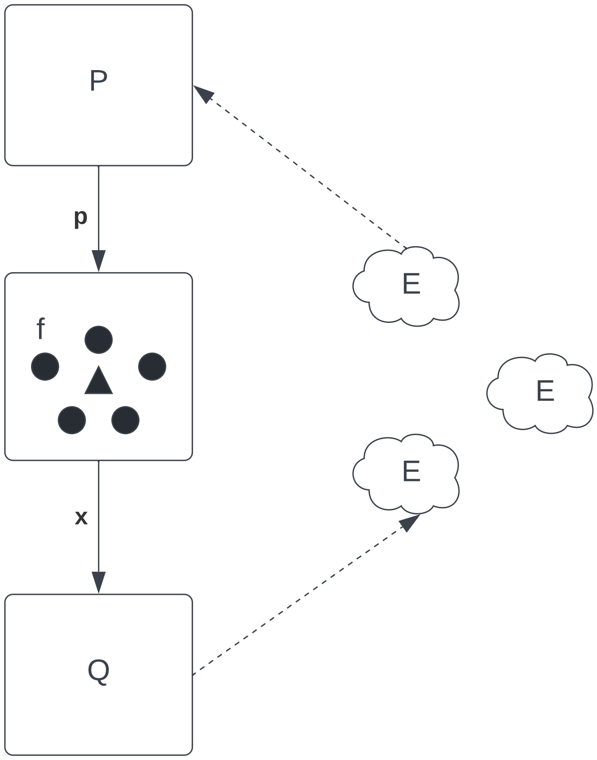


Figure 2.1: *PQf* architecture where E represents musical patterns as the environment, and ‘P Q f’ are the *analysis*, *synthesis*, and *patterning* modules correspondingly. (Blackwell, 2007)

In this architecture, Blackwell structures a Live Algorithm in three main modules: P for *analysis*, Q for *synthesis*, and f for *patterning*. These abstractions are analogous to human features as ears (P), voice (Q), and brain (f). The limits by these modules can be modeled according to perception, production and cognition within human aspects.

As noted in Figure 2.1, the system works in real-time by collecting and feeding information constantly into an environment E. The *analysis module* P ‘listens’ the environment according to the type of data stream that is captured, for instance, audio samples can be taken and analyzed to extract features such us pitch, loudness, etc. The results from the analysis process are then sent to the *patterning module* f in which the generation of new material takes place and the algorithm demonstrates its autonomy and novelty, for example, a new melody composed elementally by musical notes and durations can be created from the information extracted by P. Finally, the material should be introduced to the environment E congruently as was collected initially, in that case, the *synthesis module* Q outputs audio according to the patterns from f by using *Audio Synthesis* techniques.

This architecture is flexible enough for full-autonomy if required, however, a human operator can be part of *data injection* from the environment or *parameter adjustment* in the modules trying to set general goals but still letting the system develop its own behaviour.

The possible behaviours adopted by a Live Algorithm are: *shadowing* (follow performers synchronously), *mirroring* (style extraction to reflect back in a performance), *coupling* (behaviour driven by internal process and thus more independent from performers occasionally), and *negotiation* (like coupling but considering human cognition factors and manipulation over the performers behaviour).

These behaviours are expected to be *emergent* and resides in the f module. With this in mind, the concept of musical agents that was mentioned in previous sections fits into a Live Algorithm structure by taking actions for human-machine interactivity with respect to an environment. Agents can adopt these behaviours and demonstrate skills over constraints together with humans.

2.3 Swarm Intelligence

Following the notion of agents, *Swarm Intelligence* (SI) is inspired by nature and conceptualized as a collective behaviour of self-organized and decentralized artificial individuals (Zhang et al., 2014). These individuals are not capable of solving complex tasks as a single entity since they have limited abilities; however, when they are grouped and organized, a complex behaviour emerges to solve such tasks in mutual collaboration by transmitting information through local communication(Tan & Zheng, 2013).

Building SI systems demands flexibility and robustness. Tan & Zheng (2013) mention five characteristics that are needed to fulfill these properties which are: *scalability*, *stability*, *economical*, and *energy efficiency*. Meng et al. (2007) proposed to focus SI problems considering the features: *self-organization*, *parallelism*, and *exploitation of direct (peer-to-peer) or indirect (via the environment) local communication mechanisms*.

Blackwell (2007) applies these SI attributes to relate swarm and music considering *novelty* from a music and sound generation perspective. He defined *Swarm Music* as "a prototype of an autonomous, silicon-based improviser that could, without human intervention, participate on equal terms with the musical activity of an improvising group".

It is possible to join SI concepts and Live Algorithms to achieve a multi-agent system capable of human-machine interaction. Considering the PQf architecture presented in section 2.2, Blackwell (2007) suggests that the design of swarming music systems requires to decide the *representation* and *dynamics*. *Representation* has to do with P and Q design as well as the state of agents. *Dynamics* refers to the f module. Both type of decisions are related in the sense that a proper representation must support the dynamics in the solution created by the designer.

Blackwell claims, as a final remark in his work, that swarm simulations are a "natural choice for exploring the potential of performing machines", because of their simplicity to develop and and self-organization modelling.

2.4 Multimodal Interaction

Humans perceive and act over the world through different senses and using diverse means within their limitations. It demands the use of *modes* of communication to allow such events happen. *Multimodal Interaction* is the concept that refers to that kind of communication with virtual and physical environments through voice, body, gaze, and other modalities (Bourguet, 2003). Multiple senses can be used simultaneously for a specific interaction activity, and one modality may be supported or extended by another when communication and action take place (Haus & Pollastri, 2000) (Stivers & Sidnell, 2005).

The concept of multimodality has been used for designing solutions and analysing problems related to the human-computer interaction field. This thesis presents the design and implementation of a *multimodal system*, which is defined by Caschera et al. (2007) as "a hardware and software unit that enables receipt, interpretation and processing of input, with the integrated, coordinated production of two or more interactive modalities as output". The challenges regarding such systems rely on the integration of input modalities (fusion) and the proper output generation (fission), especially the last challenge has to be addressed as a distribution of several channels to provide consistent feedback (D'Ulizia, 2009).

To minimize the complexity and support the design of multimodal systems, Bourguet (2003) suggests the use of *Finite State Machines (FSM)* as a framework for the combination of several inputs from different modalities and the reasoning about synchronization patterns problems.

Multimodality is relevant in the music field since musical data can be expressed through visual and gestural modes, which made music representation multimodal by definition (Haus & Pollastri, 2000). In the case of a human-machine multimodal application, the machine tries to use properties from human modes to interact between its own kind and other humans as Solis et al. (2011) show in a system whose main perceptual units are vision and aural sensors that capture data in real-time to send to humanoid robots and perform music.

Developing systems under the framework of a *Live Algorithm* for a musical improvisation application can become a complex and challenging task due to the nature of music dialog required (Gifford et al., 2018). This dialog needs a proper information flow for human-machine interaction that can be framed as a multimodal system following the aspects mentioned in this section, which were applied to the work described in this thesis.

2.5 Music in Extended Realities

The previous sections in this chapter are focused on abstractions that can be used in several contexts for the conception of a multi-agent and multimodal music performance experience. One of these possible contexts is: *bringing the "merging" between physical and virtual aspects of a music improvisation*, as the one intended on this work. As such, this section presents this *"merging"* under the field of *Music in Extended Realities (Musical XR)*. Turchet et al. (2021) describe a comprehensive review and analysis in this topic that is relevant to this thesis. The main aspects of their work are presented below.

Music in Extended Realities (Musical XR) is a field that has been growing along with the technologies available in terms of *Augmented Reality (AR)*, *Augmented Virtuality (AV)*, *Virtual Reality (VR)*, and *Mixed Reality (MR)*. These technologies have opened a range of possibilities to innovate in various audio and musical contexts. Moreover, it requires an interdisciplinary effort that usually involves engineers, artists, and social scientists.

Musical XR experiences can be passive or active depending of the application, which, independently of the interaction focus, must strength the audio and music component to be considered as *Musical XR*. It means that, if the audio or music element is removed, the experience would not take place.

This emphasis given to the music component is reflected in the audio dynamic. XR experiences deal with a 3D representation of the physical world, and with the physical world itself. Thus, it is expected to have a proper spatialization of sound sources, which demands the use of *spatial audio* strategies as part of the environmental composition.

In general, XR systems explore multimodal aspects that include: *visual*, *auditory*, *haptic*, *proprioceptive*, and even *smell* and *taste* if possible. This multimodality attributes allow affordances for meaningful and engaging experiences that can enrich a musical context. Therefore, Turchet et al. (2021) take these and the previous characteristics mentioned so far to describe what *Musical XR* is. Essentially, a *Musical XR* application should comply with 4 areas explained as follows:

- **Existence of Virtual Elements:** Presentation of the elements in one or more modalities.
- **Spatial Persistence:** To employ 3D audio for spatial consistency.
- **Interactivity:** The systems should respond to users' qualities. As passive: position and head orientation; as active: manipulation of elements.
- **Sonic Organization:** It is a fundamental area that determines the system conceptualization and design according to the sound material presented.

Furthermore, Turchet et al. (2021) recommend the implementation of room-scale experiences since *"multimodality should be mapped to each other in space as closely as possible"* to improve presence, interaction, and increase affordances.

Musical XR is an early-stage field that needs research and development in the next directions:

- **Hardware:** To improve gesture interaction and feedback response.
- **Software:** To expand to specific domains in the music field.
- **Best Practices:** To define new best practices for design, implementation and evaluation of *Musical XR* systems.
- **Perception:** To put endeavors in the understanding of human perception in XR ambiences with focus on multimodality.
- **Social Experience:** To create multi-user and collaborative environments.
- **Composition and Performance:** To encourage the creation of XR music tools and the production of compositions and performances based on *Musical XR*.
- **Pedagogy:** To support teaching and learning experiences.

- **Delivery platforms:** To make platforms more accessible.
- **Intelligent Interfaces:** To improve interfaces for increasing musical engagement.
- **Standardization:** To allow the interoperability among different XR technologies.

Part of the basis of this thesis relies on the concepts and frameworks from *Musical XR*, since it facilitates the creation of experiences that are not found or are difficult to reproduce in the real world, as the one reflected on the proposed system described in this manuscript.

2.6 Related work

The system this thesis focuses on, integrates several concepts, paradigms, and technologies that have been used for a variety of musical applications explored individually or in combination. The abstractions and strategies presented in the previous sections have led to the design and implementation of systems that combines audio and visuals in real-time environments regarding music performances.

Such is the case of *REVIVE* (Tatar et al., 2018) (Tatar et al., 2019), which uses these concepts for a live interaction with musical agents, human musicians, and visual generation agents through the *Musical Agent based on Self-Organizing Maps (MASOM)*, a *machine listening* strategy that collects data (from artificial agents and human musicians) in real-time to decide the next musical material in the performance. A similar approach is adopted by Bown et al. (2015), who presents *Musebots*, a platform where autonomous agents create music among themselves, as well as *Spire Muse* by Thelle & Pasquier (2021), a system also based on MASOM whose interface enables musical agents attributes as described in section 2.1.

Agents are usually grouped to develop emergent behaviors that lead to *swarm intelligence* as illustrated in section 2.3. For swarm systems that foster human-machine interaction, the concept of *reciprocity* explored by Choi (2018) establishes a relationship between the human performer and evolutionary swarm tendencies for music generation. He demonstrated, by implementing two musical works, that reciprocity leverages emergency and contributes to the shared control among participants of a music performance.

This concept of *reciprocity* can be seen as a relevant characteristic in improvisational systems as the ones described by Gifford et al. (2018) such as: *Music Mouse*, a rule-based system that affords interaction just with a mouse and a computer screen; *Cypher*, a machine listening system that can interact with another human musician or generate material by itself; *Voyager*, an improvising orchestra that is considered the first in using multiple players (agents) as improvisers and allows to choose among several algorithms; *JITLib*, a live-coding framework; *Shimon*, an animatronic machine improviser, *Wekinator*, a machine learning framework for real-time music control; and the *Reflexive Looper*, which generates music material based on the style of a human performer in real-time with a multimodal approach. These works are considered base examples for describing improvisational properties that are mentioned in section 2.2.

By the other hand, works that implement strategies under the concepts described so far, needs to prompt accessibility to users through their senses. In this case, multimodal solutions can be achieved through technological approaches that are of interest to this thesis, which are: *Motion Capture (MoCap)*, *Spatial Audio*, and *Extended Reality (XR)*. They are usually together or with other technologies according to the application.

For motion capture, this work considers an *Optical MoCap System*, which is a platform composed of infrared cameras that shapes a 3D space and tracks reflective markers as points in a virtual environment. Costagliola (2018) used such system for a multi-user augmented audio application, in which he tracked the position of people's heads to stream a binaural sound that was panning according to the individual place of a person. He managed to have a shared realistic environment with low-latency and high precision. Müller et al. (2014) developed *The BoomRoom*, which allows a person to *"touch", grab and manipulate sounds in mid-air*, they used real objects for visual feedback and an optical motion capture for gestures and objects position. They studied the localization accuracy for a *Wave Field Synthesis (WFS)* spatial audio setup.

There are other relevant works that have employed *spatial audio* as the focus of their research. Grani et al. (2015) explored spatial audio in virtual environments and found that audio-visual attractors are the most efficient to capture users' attention. Robinson (2020) presented a prototype for human-robot interaction that uses spatial audio for environmental communication.

There are representative works related with *Extended Reality (XR)* that fit into the field of *Musical XR* explained in section 2.5. Hamilton et al. (2011) describe multimodal music environments that use mixed reality. Some of these spaces deal with networked music and metronomic pieces, as well as the use of spatial audio for geolocalization in virtual spaces, moreover, part of these environments map spatial coordinates for music control and sampling.

As part of the *Extended Reality (XR)* technologies, one of the most advanced devices in mixed reality (MR) is the *Microsoft HoloLens*[1], which is a MR headset that performs a precise mapping of the physical space to place virtual augmentations. Hockett & Ingleby (2016) present several cases for mixed reality visualizations for this device. They claim that the HoloLens fulfill the criteria of "being immersed in the data" despite of its limitations, and recommend this device as a tool for advance AR/MR applications in different fields.

The HoloLens has been used in a variety of applications such as the work presented by Das et al. (2017), which is an augmented reality piano learning system that combines the headset with Bluetooth-over-MIDI communication and hand tracking. It is also possible to map virtual objects with sound properties and interact with them as Nakagawa et al. (2018) show in their work. They implemented a system capable of changing pitch and timbre based on objects position. Selfridge & Barthet (2019) used the headset for a study with an audience to compare the usefulness of the MR device against mobile devices when emotions (represented as emoticons) were triggered from the same audience while listening a live performance.

Considering MR through the HoloLens as a way for music performances in a human-machine collaboration, Riley (2021) presents a set of three MR software applications for music-making developed on HoloLens 2. Although these applications do not have sophisticated music generation algorithms, they explore affordances regarding the merging between the physical and virtual world for live performances. The main features of these applications are related with multi-track instrument mixing, inspirational visual landscapes, and interaction among virtual objects and the physical space in terms of collision and sound mapping.

The work that is closer to the proposed system in this thesis is the one presented by Bullock et al. (2016). In terms of conceptualization and prototyping, they portrait an interface for representing sound sources as visual objects that can be manipulated in a 3D space. This objects have physical properties regarding the environment and can be visualized through a screen monitor, as well as listened in a 3D spatial audio system (binaural or loudspeakers array). They argue that "direct manipulation aspects are, within the context of interfaces for audio transformation, key indicators of a highly positive user experience".

Certainly, there are more research projects from the vast literature that are related. Some of them are not included but were examined to find key aspects for possible contributions to the theoretical framework. From this search, the author has not

[1] https://www.microsoft.com/en-us/hololens

found a similar system as the one described in this manuscript. However, all the presented works provide guidelines and motivation for the conception of this work.

2.7 Summary

This chapter presents a theoretical framework in terms of key concepts and related work for aspects on *agency for human-machine music interaction*, *multimodality*, and *music in extended realities.*

This research fields are supported by cutting-edge technologies that try to maximize the immersive experiences regarding audio generation and music-making by involving the merging between the physical world and virtual augmentations. Particularly, *motion capture*, *spatial audio*, and *extended reality (XR)* platforms are the preferred tools to build complex interactive systems that may involve individual or collective user interactions.

The works reviewed in this chapter show the use of abstractions and implementations for solutions that exploit the capabilities of the technologies mentioned above and paradigms for music performances. However, after a deeper exploration of the literature, no system has yet been proposed that integrates *motion capture*, *spatial audio*, and *extended reality (XR)* technologies in one only integrated human-machine system for music improvisation. Moreover, there are no evaluations for similar systems in terms of parameters that may potentially affect the user experience, as well as music performance sessions to describe the human-machine behaviour that emerges in the interaction.

This gap described in this chapter further support the development of a novel interactive music system that involves these vanguard technologies and bases its design on the abstractions and strategies related to *Musical Agents*, *Live Algorithms*, *Swarm Intelligence*, and *Musical XR* shown in this chapter.

Chapter 3

Methodology

This chapter presents the methodology for evaluating the proposed system detailed in Chapter 4. It illustrates the procedure followed in this thesis to answer the research question, as well as strategies to carry out measurements over the system implementation and user testing. An overview of the system design is provided in order to demystify the materials and methods adopted in this work.

3.1 Research Procedure

The idealization of the system in question was conceived under the vision of the author for bringing a novel platform for human-machine music-making. As starting point, recent applications and fundamental paradigms — described in Chapter 2 — were explored to support and modify the idea.

As part of that exploration, the available technological resources on the *MCT Portal*[1], located at the *Department of Musicology* (*University of Oslo*), were used for experimenting independent modules required for the system development. The experiments and projects implemented in this laboratory gave lights to establish advantages and limitations in terms of: *audio systems* (included a spatial audio setup), *motion capture*, *physical space*, and *computational power*. The mixed reality device, the *Microsoft HoloLens* (version 1), was provided by the *Norway University Hospital – Rikshospitale* where research regarding medical applications on Extended Realities (XR) is conducted[2].

The system was designed and implement by following an iterative process focused on the integration of several modules. This modules were implemented by the author

[1] https://www.hf.uio.no/imv/english/about/rooms-and-equipment/mct-portal/

[2] https://oslo-universitetssykehus.no/om-oss/nyheter/holoviz-fra-2d-til-3d

and other were integrated from open-source third-party resources. The process and the system itself needed to be evaluated to validate that the way-of-making fulfills the objectives established in the general vision.

Essentially, the solution is an *Interactive Music Systems (IMS)* under the category of *Digital Musical Instruments (DMIs).* O'Modhrain (2011) describes a framework for the evaluation of DMIs where the methodologies depend on the stakeholder. He recommends to have a clear understanding of what apply and to who depending on the interest of the study. For this system, the stakeholders are the *performer/composer*, and the *designer* (the author of this thesis).

From that perspective, the designer is interested in the *effectiveness* and *efficiency* of the system, as well as the *effects of the design decisions* over potential music performers. Therefore, the proposed evaluation follows the next strategy:

1. **Stress tests to figure out relevant limitations**, i.e., the number of agents to be used.

2. **Data collection of system measurements:** Latency, packet loss, and real-time parameters.

3. **User testing - music performance: Data collection from events generated in the system**, e.g, recording of objects' positions, controller manipulations, audio output, etc.

4. **User testing - music Performance: Survey and reflections** from a musical improvisation using the system.

5. **Data analysis** for systems measurements and user recordings.

6. **Comparisons and reflections** from the testing to criticize the system and its making process.

The next sections give specific details in how to carried out this evaluation process in terms of the data of interest.

3.2 System Design

The design of the system described in Chapter 4 is supported by three cutting-edge technologies (Motion Capture, Spatial audio, and Mixed Reality) and a musical input source through a MIDI controller. These components are connected to the core of the solution, which means that there are five elements that assemble the complete platform as shown in Figure 3.1. Remember that an *agent* is the representation of a *sound source* located in space that contains musical material from the performer and is able to adopt an autonomous behaviour when the user decides it.

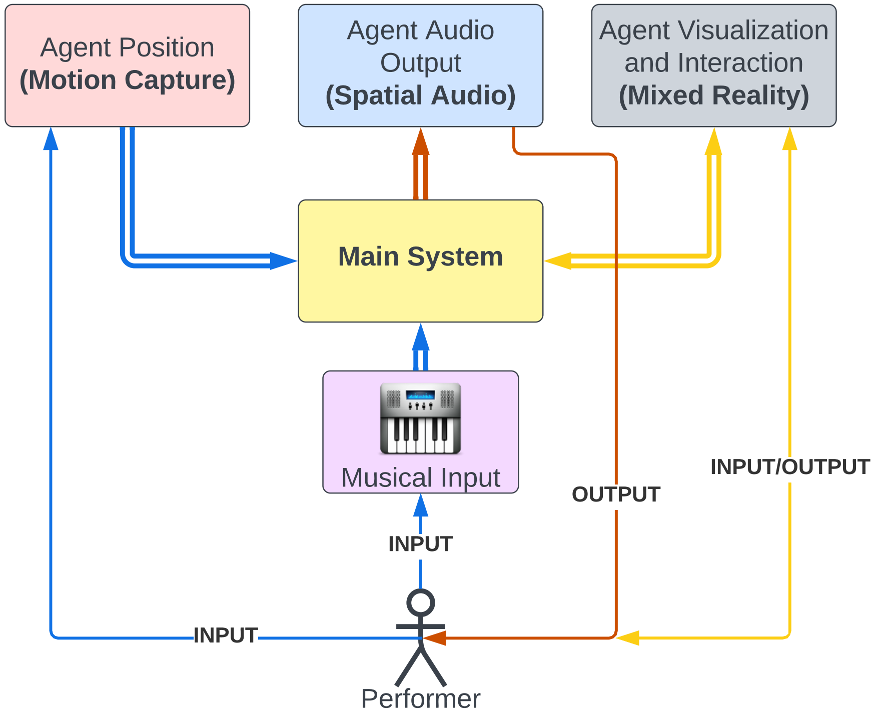


Figure 3.1: *System Design Overview.* The wider arrows represent inputs and outputs from the component to the system and the thinner ones show inputs provided by the user and outputs received from the system.

The technologies illustrated in 3.1 allow the user to interact with the system and receive feedback in real-time. An *agent* (sound source) is initially an audio track that is created when the human performer finishes a musical line that is input from the MIDI controller and looped it. The sound source can be moved in space through a trackable object called *rigid body* whose position is detected by an *optical motion capture system*. The position is fed into the system to place the agent in space in terms of sound and image. For the audio output, a *spatial audio system* composed of a circular array of loudspeakers reflects the location of the sound source. For visualization, a *mixed reality* headset is used to render the agent (as a 3D colored sphere) in the physical space where the performer is, also the headset allows to interact with simple gestures to select the agent to control it or let it to behave autonomously, that is why the mixed reality device receives and sends data to the main system.

The communication between components depends of the technology that is using. For the spatial audio system there is an audio infrastructure described in section 4.3.3. The motion capture system and the mixed reality headset are connected through a local network (LAN) over a router. The motion capture uses a wired (Ethernet) connection while the mixed reality headset uses the wireless network.

These interconnections have an impact in the size, time, and rate of data transmission back and forth, which is significant for the user experience in a real-time setting. That is why it is relevant to take system measurements as part of the methodology to evaluate this system, since those measurements can help to improve and find a balance between the experience and the system efficiency, or identify the feasibility for music performances under these limitations (Schuett, 2002).

For a detailed explanation of the complete system refer to Chapter 4.

3.3 Number of Agents

The system allows to configure a specific number of agents to interact with. However, because of resources limitations such as computational power and network throughput, the number of agents is finite to the point in which the quality of the audio output and the user experience is not significantly affected.

As such, in order to choose the right number of agents for the evaluation proposed in this chapter, the system has to be tested on the hardware and software platform where is running to find that value. For this, the strategy proposed is *adjusting the sample rate and the buffer size of the audio device and the software component until dropouts are found in the audio signal.*

The dropouts leads to audio artifacts that affects the quality of the output. Moreover, the latency between the musical input and the audio output increases when the sample rate is reduced and the buffer size is increased, but it avoids dropouts. Therefore, it is important to find a trade-off to balance these parameters.

The rest of the evaluation is applied per each set of agents up to the value found in this stage.

3.4 System Measurements

The most relevant and critical attributes in a real-time digital music system, as the one proposed, are *latency* and *jitter* (variation of latency). Several latency categories that involve action and perception from users are measured to identify *how* and *if* it changes and affects the efficiency as the number of agents increases.

As there are components that are communicated through network interfaces, an additional parameter to measure, apart form latency, is the *packet loss*. If there is a significant amount of data that does not arrive to the destination, it could affect the reactivity of the system when commands are transmitted, or produce gaps for continuous data used for updating real-time virtual objects. As with latency, this measurement might be affected according to the amount of agents since the data packet to transmit is longer as they increase.

Other measurement methods concerning to parameters related to the real-time functioning of the system are presented. They can impact in the comprehensive efficiency and user experience depending on the chosen solutions. That is, the processing of the algorithms for autonomous movement and music generation, as well as other calculations, are measured in terms of computational time, which contributes to corresponding latency values.

The methods for performing these measurements are explained in the following subsections.

3.4.1 Latency and Jitter

The previous diagram in Figure 3.1 shows the connection between components in the system. From one component to another exists a delay when it is expected immediate reaction from certain events. This delay is the latency between those components, which can possibly has variation (jitter) that should be identified from the measurements.

For determining these parameters per each number of agents, several methods are shown in the next parts.

MIDI Controller to Sound Output Latency

The latency between the moment in which the performer *press a key in the MIDI controller* and the *sound output is heard* is measured as Figure 3.2 suggests.

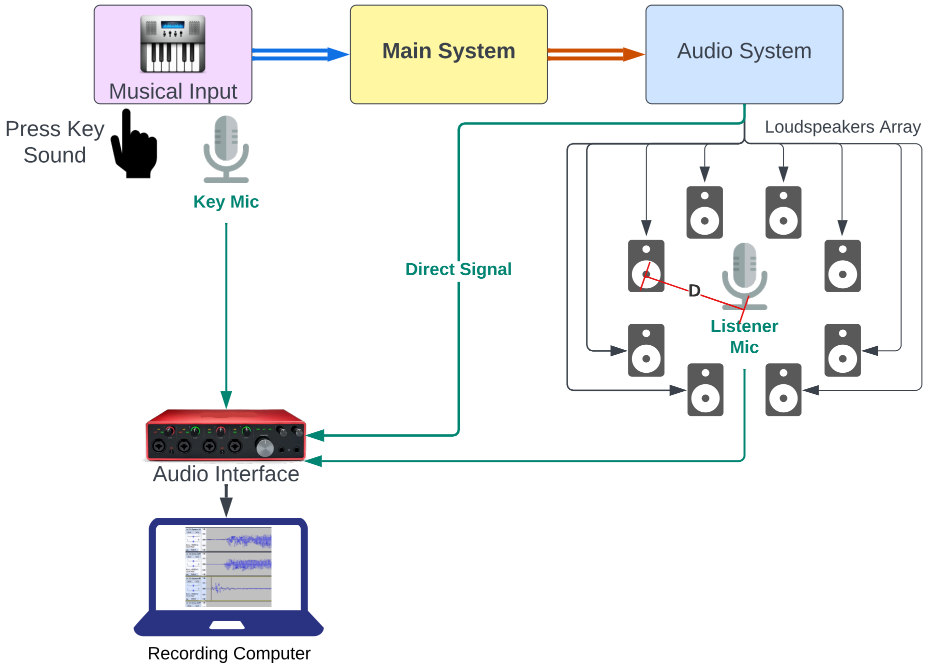


Figure 3.2: *MIDI Controller to Sound Output latency measurement method.* Three signals are recorded to figure out the time in which the audio output travels from the instant when a key is pressed to the listener

In this method, three audio signals are recorded with an external equipment to figure out two latency values of interest. The first signal comes from a microphone that captures the sound of pressing a key in the MIDI controller, the second one is the direct sound from the audio system, which is the same signal that feeds the loudspeakers array for the spatial audio setup, and the third one is the signal recorded from a microphone in the center of the loudspeakers array that can be considered as the average point in which the user will move. Note that this point is placed from the loudspeakers by a radius of D units.

The two values to measure are the latency between the *key when is pressed* to the *direct sound*, and from the *direct sound* to the *listener* through the loudspeakers. It is expected that the second value is approximately equal to the time in which the sound travels on air in a D distance. An audio editor es needed to inspect the signals and identify the time difference between them.

The experiment consists in recording a set of 30 or more key pressing events per each size of agents until the maximum number of agents chosen. On every run, the sound source that is tested is recorded while the other sources are still playing but with zero volume to capture only the last source.

The total latency is the sum of the two measured values and the jitter is the standard deviation of the data set collected for every agent size.

Spatial Audio Placement Latency

The user is able to move a sound source (agent) in space through a motion capture system. The object that is tracked is called *rigid body* in the terminology of an optical motion capture platform. This rigid body represents a point in space with (x, y, z) position coordinates and (pitch, roll, yaw) rotation angles. The agent is virtually attached to this body so that the sound location and the visual feedback follow it.

There is a time between *the instant when the rigid body reaches certain position in space* and *when the spatial audio system places the sound in the loudspeakers array.* This latency is measured in two steps: first, the delay between the physical object (rigid body) and the main system; and second, the delay between the main system and the spatial audio placement. Note that the overall latency includes the time that takes the sound to reach the listener, which is measured with the method presented in the previous section. Therefore, the total latency is the sum of the result from the two steps described before plus the latency between the speakers and the listener.

As the first step, for measuring the **latency between the *rigid body* and the *main system***, it is proposed the method depicted in Figure 3.3.

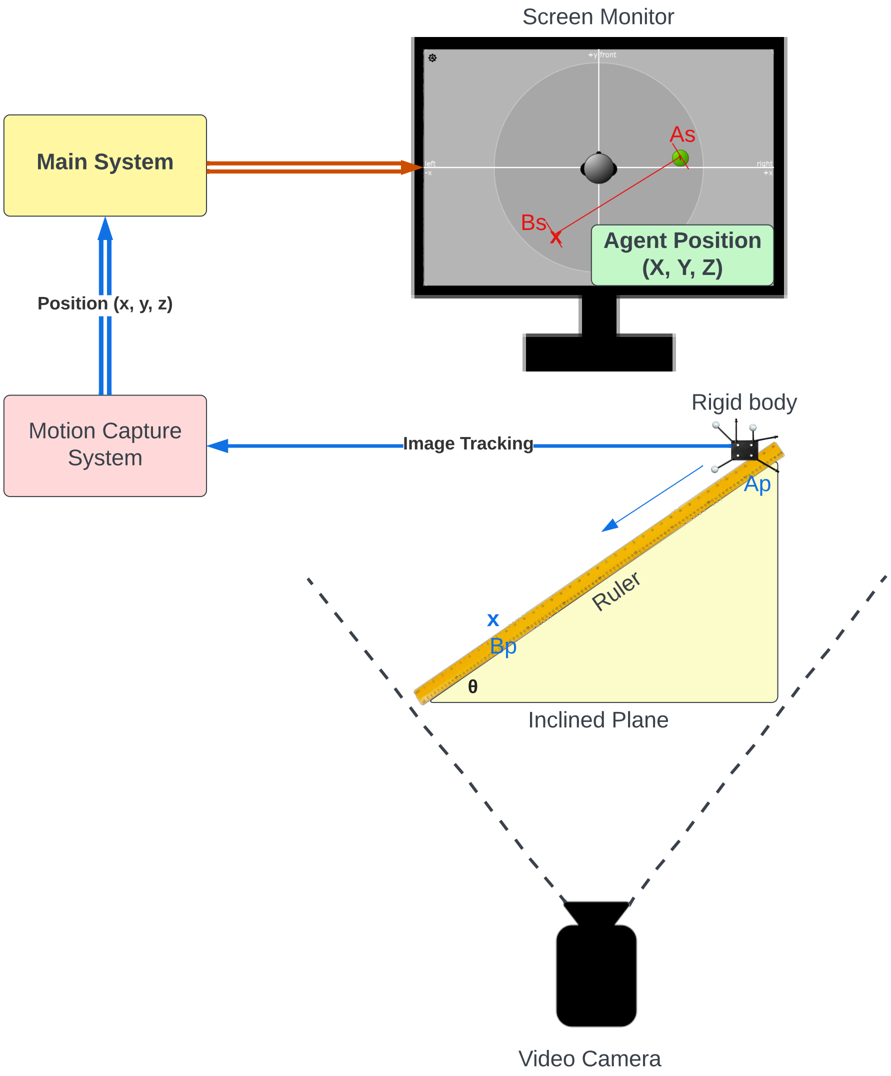


Figure 3.3: *Rigid body to Main System latency measurement method.* A video camera records the physical movement of the object and its tracked position received from the motion capture to determine the difference in distance and time.

The strategy consists in using an inclined plane of angle θ to let the rigid body move freely from a point Ap to a point Bp as illustrated in Figure 3.3. While this is happening, a monitor screen shows the object coordinates that the main system receives from the motion capture. The screen displays the movement from a point As to Bs. The inclined plane has a ruler attached in the direction of the movement so that a video camera with high frame rate can record both, the value in the ruler and the coordinates in the screen at the same time.

Using a video editor, it should be possible to inspect at any time the position of the rigid body in the ruler and in the screen. With that information, we can calculate the traveled distance physically (Dp) and virtually (Ds). For the ruler we subtract the initial point Ap — in which the body starts to move — from the point Bp that is taken later in the movement, for the case of the screen we use the *euclidean distance* between those two same points represented as As and Bs. It will be noticeable that the distance from the ruler (Dp) is greater than the one registered in the screen (Ds) because of the latency, thus finally we find in the video editor the moment in which the object arrives to the virtual distance Ds in the ruler. The difference in time between those two moments in the video is the latency of interest.

The inclined plane of angle θ can be chosen according to how fast we want the body to travel. Consider that the friction between the object — or something that transport it — and the plane affect the movement. The recommendation is keeping an angle that is good enough for recording. If the video camera has a low frame rate we would need low speeds for better capturing.

The advantage of an inclined plane is that the speed increases as the object moves, which can be useful if we want to find the latency at different speeds.

An important consideration is that the latency between the system and the rendering of the position in the monitor screen must be subtracted from the measured latency. This value can be found in the technical sheet of the screen.

For the second step, **the latency between the *main system* and the *spatial audio placement* in the loudspeakers array** is measured following the technique shown in Figure 3.4.

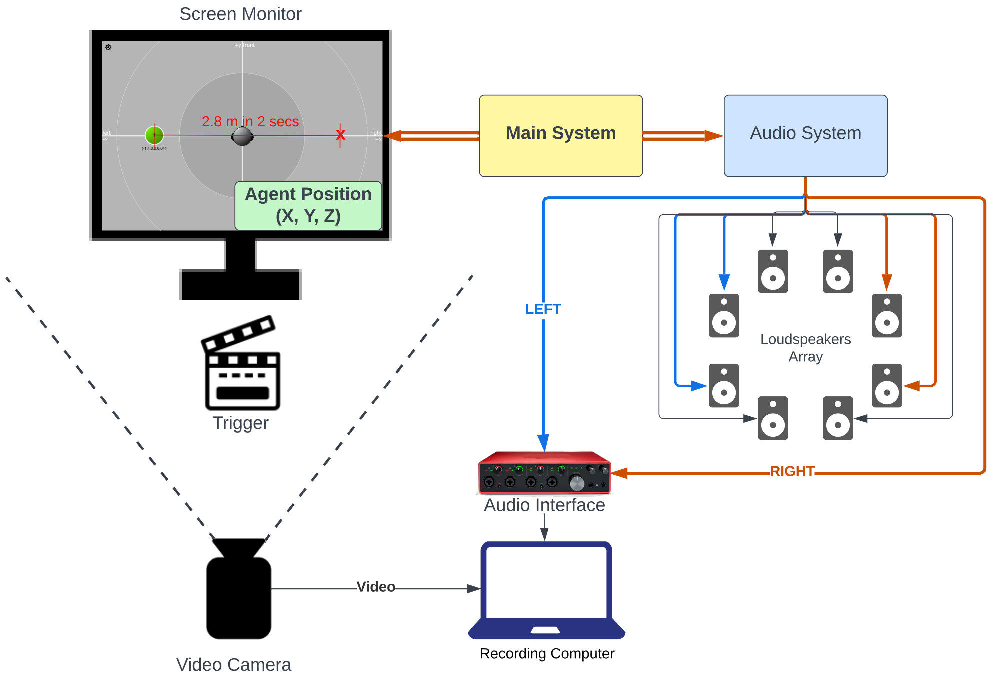


Figure 3.4: *Main System to Spatial Placement latency measurement method.* A video camera records a rendered version of the agent — moving left-right constantly — from the main system while capturing the left and right signal sources from the audio system. A trigger must be used to synchronize audio and video.

In this case, the main system move a sound source (agent) constantly from left to right at 1.4 m/s, which can be considered as an approximation of the average walking speed for humans. In that sense, if the object move in 2 seconds from one point to the other, it will cover 2.8 meters. This rendered version of the sound source, along with the position coordinates, is displayed in a monitor screen which is captured by a high-frame-rate video camera.

Additionally, it is recorded a constant playing sound wave that changes its amplitude uniformly according to the movement, which goes from the left to the right side of the loudspeakers array. Since the sound is recorded independently from the video, it is needed to use a trigger cue — such as a clapperboard — to synchronize audio and image in a video editor.

For taking correctly the left and right side of the loudspeakers, depending on the configuration, the signals to take can be one or two depending of the number of speakers and their position. If we assume positions as the ones shown in Figure 3.4, the two adjacent speakers to each side must be taken and mixed as one for an even signal. The capture and mixing of those signals depends on the audio system that is being used, that is, if it contains a digital mixer, it should be possible to take more outputs where those signals are sent without modifying the speakers setup and connection.

A sine wave is used to avoid shapes in the waveform that could affect the observations. Figure 3.5 shows an example of a resulting waveform from a recording of the left-right movement.

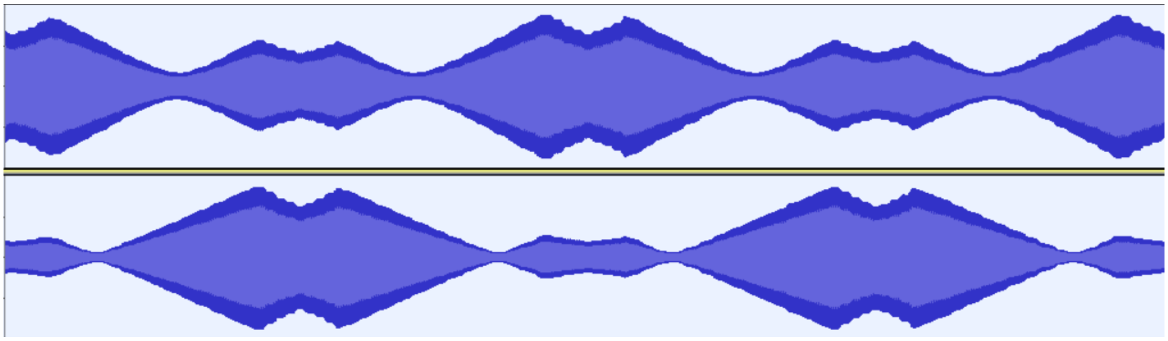


Figure 3.5: *Sound wave from the resulting movement.* The left and right sides of the loudspeakers array is recorded to identify differences in the position of the source in the video regarding the waveform shape.

For determine the latency, the audio and video are merged in a video editor and synchronized through the trigger cue, then the left or most right position of the sound source in the video should be identified. If, for instance the right extreme is chosen, we look at the sound waveform for the highest pick in the right channel, which should be the right extreme in that moment for the audio. We will notice that the two points fall in distinct moments in time, hence the difference of those timestamps is the target latency.

In Figure 3.5, the waveform suggests that the extreme is actually between the two highest points. This is because a modification in amplitude happens when the sound source is moving away from the center, similar to the physical world in which sounds are attenuated as the distance increase.

For data collection, the recording could last until 30 or more cycles and per each agent size, which means that the amplitude of the other sound sources should be reduced to zero to hear the source of interest on every run. Having samples per each agent size will help to determine if the latency changes significantly when more sound sources are added.

Sound to Visualization Latency

At the moment we hear a sound source (agent) located in one section of the speaker array, it is expected to confirm its position visually (as a 3D sphere in space). It is done in the implementation through a *mixed reality headset*. However, there is a delay between sound and image that would produce a visual misplacement of the source.

The mixed reality headset is capable of rendering more than one agent at the same time through its graphics engine. Thus, this device receives constantly a packet that contains the data for all agents over a wireless local network, that is, the packet size increases as the amount of agents rises, which could also increase the sound-visual delay.

To measure this latency, since there are more steps involved from the moment a packet is sent to the final render, it is necessary to know the time it takes to draw the agents at different stages and measurements methods, which are presented as follows:

- **From the *main system* to the *“packet receiving module” in the mixed reality application*:** The time from the moment a packet is sent by the main system and received by the mixed reality application is the *network latency*, which is measured approximately as the half of the round-trip time that a packet takes. It implies to include an additional module in the mixed reality application to send back the packages, and another one in the system to take a packet back and identify the timestamps when sending and receiving. Figure 3.6 depicts a basic diagram for this process. This time is included in Figure 3.7, which illustrates several latency stages in a timeline.

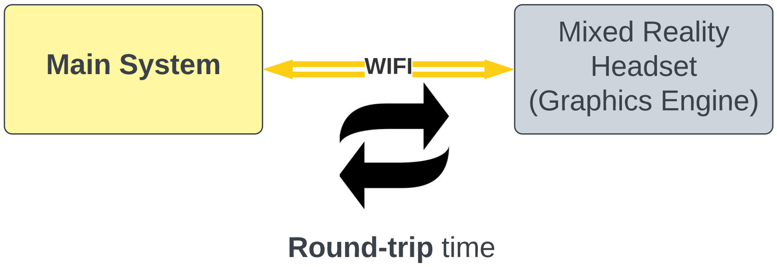


Figure 3.6: *Round-trip measurement latency.* The main system sends a packet and receives it back from the mixed reality application in order to measure the time it takes to travel through the network.

- **Inside the mixed reality application: from the "packet receiving module" to "position update":** Since a packet is received in a different thread that the real-time processing thread (where the position for the graphical object is set), there is a time between this two points, which sometimes could be as much as the time of the *frame to frame render process* or almost immediate if the packet arrives just before the frame starts. This time is measured inside the mixed reality application and does not have to do with the main system. Figure 3.7 shows this latency as the distance between t_r and t_p.
- **Inside the mixed reality application: The "real-time application loop":** There is a time between frames to update the mixed reality application since it works under a game engine infrastructure. For including this delay as part of the overall latency, the frame to frame time difference in the application loop is measured. As the positions of agents are updated on every frame, the frame to frame time can be reflected as the difference between tp and tg as shown in Figure 3.7.

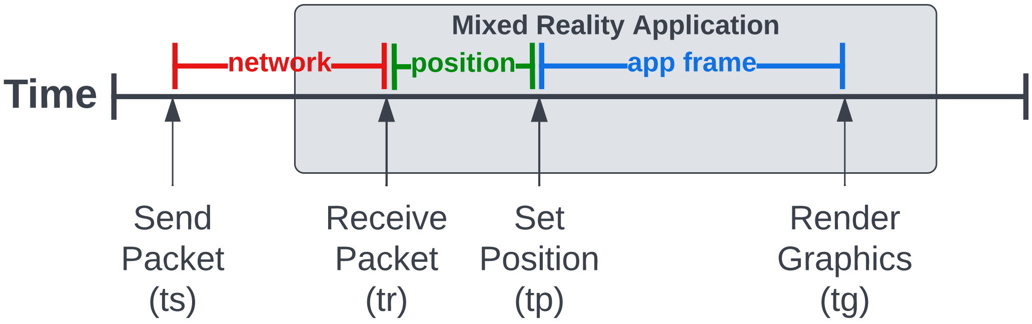


Figure 3.7: *Main System to Mixed Reality Application packet timeline.* These points reflects the moments in which a packet with position information is rendered in the mixed reality device.

In the implementation, when a position is received, it actually sets a *target position* where the agent is supposed to be. Thus, from the last position received to this target position, there is an interpolation process when the time between points is greater than the frame time. Additionally, there is a *smoothing* process to visualize a continuous movement. It is required a time to perform these processes, which is assigned by the developer in the mixed reality application. This time has to be summed to the overall latency.

The latency resulting from the sum of the previous stages is from the *main system* to the *visualization*. To get the latency from the *audio output* to the *visualization*, the *spatial audio placement latency* — explained in the previous section — has to be subtracted.

The experiments collects 30 or more samples per each stage and for every agent size.

3.4.2 CPU Usage

The CPU usage is the percentage of work carried out by the processor regarding its total capacity. To determine the impact in computational processing per agent, this percentage is measured by taking 30 or more samples separated by a small period during a certain amount of time. The framework for the implementation (Max/MSP/Jitter) provides the tools for this measurement. The outcomes should show how this percentage changes in time.

3.4.3 Packet Loss

There are two components connected to the main system through a local network: the motion capture (Ethernet), and the mixed reality device (WIFI). Since these components are using *OSC* messages to communicate each other, there is no confirmation of message reception, hence the communication is prone to losing packets along the way.

The packet loss represents the percentage of packets that did not arrived to its destination. To calculate this value, it is needed the number of packets from the *component that transmits* P_s, and the number of packets from the *component that receives* P_r over a period of time. The packet loss P_{loss} is given by (3.1).

$$P_{loss} = 100\frac{P_s - P_r}{P_s} \tag{3.1}$$

The proposed measurement method is presented in Figure 3.8. Additional modules needs to be implemented on each component to count the packages and inform each other about when to start and finish the counting. The strategy consists in sending a command to *START* the count from one module to the other during one minute, then sending a command to *END* the count. Once the external module receive the *END* signal, it requests to start a new repetition for another minute. In that way, it is possible to configure the number of repetitions (30 or more) to have a data set for each agent size. This procedure is performed to both, motion capture system and mixed reality device.

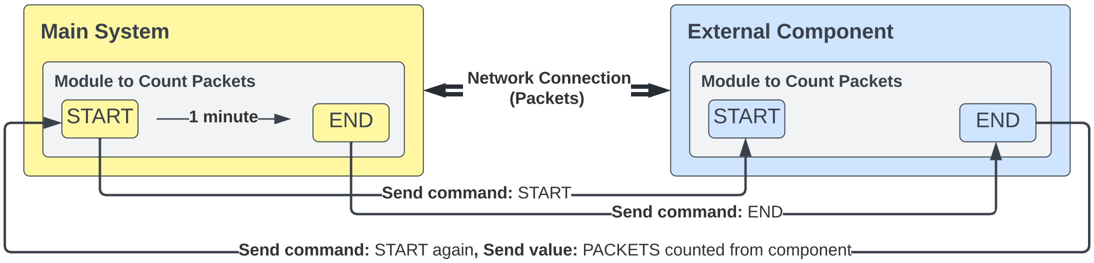


Figure 3.8: *Packet Loss measurement method.* The packets sent to or from modules connected through the network are counted for one minute over a certain number of repetitions (30 or more) to get the amount of sent and received packets, then the loss percentage is calculated.

3.4.4 Autonomous Algorithms Measurements

The algorithms for autonomous agents' behaviour require computational resources and processing time. That is why it is relevant to measure the impact of these strategies by following the methods proposed below.

Music Generation Algorithm

The chosen algorithm for music generation is a framework of several *Markov Chains* modules as explained in section 4.2.9. To figure out the impact of one of this modules in terms of computational time, the following two cases are evaluated.

- **Build time:** The user inputs musical material through a MIDI controller to the system for feeding an agent. This input is taken by the Markov Chains modules and then they are trained when the agent is *released* (i.e., the autonomous behaviour is activated). The training process *builds* a model for each module, which incurs in the use of a certain calculation time. To measure this time in several cases, an experiment that considers 30 build cycles is carried out. Each cycle contains a *training samples size* from 100 to 3000 with steps of 100. The samples are random numbers between 1 to 120. The building time is measured 30 times for every training samples size and per each agent size, that is, up to 900 measurements per each agent size. A module for an automatic running of this experiment is implemented in the main system.

- **Generation Time:** Once the model for every experiment regarding *training samples size* is built, the generation time for a new sample is measured per each agent size by taking the difference between the moment when a new sample is requested and when the output is provided. Another module for performing this measurement is implemented and run to acquired 30 samples or more on each cycle.

Movement Algorithm

The autonomous movement algorithm for the agents, as an artificial swarm, is computed in the main system through a continuous task in a real-time context. This task is scheduled to be executed in a fixed frame time (30 ms as explained in section 4.2.9). Thus, it is relevant to measure this time in the implementation per each agent size and figure out the actual value when the full system is working.

An additional measurement is the *processing time of the task* that represents the execution time of the movement algorithm. It is also captured per agent size.

This data is recorded during one minute for each case.

3.5 User-Agents Interaction

For exploring the user experience regarding the human-machine interaction, a set of real-time data is recorded from several participants. The captured data is anonymous and does not provide any indication of the identity of the participant. Furthermore, the participants have a minimal background to play at least a musical keyboard and improvise.

The parameters that are recorded are the following ones:

- MIDI controller events (note on, note off, velocity, and control change — used for looping control and sound synthesis parameters).
- Metronome beats.
- Rigid body position (from the motion capture system).
- User position and gaze direction (from the mixed reality device).
- User interaction for releasing and locking agents (from the mixed reality device).
- Agents position (from the main system).
- If agents are in the users' field of view (from the mixed reality device).

The implementation of a recording module is required to capture the data listed above. This recording module is attached to the main system so that it runs while a participant is using it in a music performance session.

The data collected might be analyzed and presented in several ways, however, because of resources and time restrictions, this thesis considered only the following aspects:

- **Heat maps** to identify frequent locations for the *rigid body*, *user*, and *agents*; as well as how they relate each other in terms of distances between those elements, which are illustrated as **Distance vs. Time graphs**.
- **The amount of musical material** that the users inputs into the system along the time.
- **The agent manipulation**, it means when the user activates (release) or deactivates (lock) the autonomous behaviour of an agent during the session.
- **The user attention**, which is determined by the the gaze direction during the music performance, as well as the amount of agents found in the field of view.

There are no formal expectations about the results from this data analysis. The goal is the identification of aspects that are discussed for further recommendations, improvements, and future studies in the human-computer interaction field regarding this system.

Additionally, the participants filled a survey shown in Appendix B. This survey, as well as critical reflections, are analyzed qualitatively since the number of participants is not enough for a quantitative processing from the closed-ended questions.

3.6 Summary

This chapter presents several methods for measuring relevant parameters for the system proposed in this thesis, as well as means to capture data regarding user experience.

For having statistically significant results, the corresponding methods suggests to collect 30 or more samples. The results in Chapter 5 reflects this suggestion and presents a series of *quantitative descriptive analysis* — and *inferences* where it is needed — according to the data collected from the *system measurements*.

User-agents interaction data is summarized in various graphs illustrated in Chapter 5 and interpreted in Chapter 6 as a critical discussion of the results.

The system evaluated by the methods presented above is described fully in the next Chapter.

Chapter 4

System Description

This chapter presents the design and implementation of a system for human-machine music improvisations. This system is based on a multimodal approach that involves *Visual*, *Auditory*, *Haptic*, and *Proprioceptive* sensory modalities. The machine aspect is modeled as a *Multi-Agent System* that prompts behaviours in terms of *music generation* and *autonomous movement*. The implementation considers three cutting-edge technologies which are *Motion Capture*, *Spatial Audio*, and *Mixed reality (MR)* that were integrated in such a way that it delivers the best performance possible.

In the first place, this chapter lists the requirements that define the design and implementation directions. Then, the design is described in terms of abstractions and strategies illustrated in Chapter 2, and from other specific frameworks. Finally, concrete platforms, tools, and integration procedures are detailed to show how the final implementation was carried out.

4.1 Requirements

4.1.1 Overall Description

This thesis proposes an Interactive Music System (IMS) that allows a musician to improvise music together with artificial entities. This entities are represented by *sound sources* that can be manipulated in space by the human musician, that is, the performer is able to *move*, *hear*, and *see* them in the physical place where the improvisation is happening.

This *sound sources* are created through a *looper* as in a normal *multi-track looping session*. This looper is synchronized with a *metronome* at a specific musical time that can be changed by the user if need. Moreover, the user has a musical keyboard that is used as the interface for playing musical notes (piano keys) and changing synthesis parameters for the sound (knobs). In essence, the performer is able to record a music line of a particular sound and loop it, then play on top of the loop and continue changing the synthesis parameters in real-time. This is a traditional process for a looping session in which normally, at some point, the musician decides to add another track through the looper capabilities and start to build a multi-track composition.

In the case of the proposed application, while a music track (sound source) is recorded — or even after — , the user is able to manipulate its position in space by using a third physical object. Let us name this object as the *"Spatial Positioner"*, which is something that can be easily moved with one hand and placed anywhere the user wants. The purpose of the *spatial positioner* is serving as a tool for moving the sound source in space, which means that, where this object is, the musician can hear that his or her sound source is coming from that position. Additionally, the user should be able to *visualize* the sound source over the *spatial positioner* so that he or she confirms its presence in space visually and audibly. One of the simplest shapes that can represent a sound source in space is a *sphere*. Thus, the musician can identified the music track (sound source) in the room as a sphere on top of the *spatial positioner*, wherever it is.

The process described above involves the interaction with one only sound source, which just obeys to the actions executed by the user. However, this sound source is a potential *artificial musician* that can play along with the human performer when he or she decides. Before this decision, this sound source is *"locked"* to the will of the musician as explained before, and it is fed with the musical material that the user input when the sound source was recorded through the looper. It means that the sound source is *trained* with the music line from the human musician in terms of musical notes from the keyboard. Let us identify now this sound source as an *"Agent"*.

In order to enable the autonomous behaviour of an agent, it has to be *"released"*. For doing this, the user can just point to the sphere with either hand and select it remotely with an *air tapping* gesture targeted to the object. In that moment the agent will detach from the *spatial positioner* and start to move freely in the room, moreover, it will change the musical material that originally was recorded in the loop, that is, it will play a different musical line to its will but trying to keep the playing style with which was originally recorded, as well as the same musical tempo.

When this first agent is *released*, the musician will note that a new sphere is placed in the *spatial positioner*. This is a new track (agent) that waits for being recorded, and the same interaction will happen as the first agent (record loop, move the *spatial positioner* if needed, and release it). This second agent may have a different sound and musical line according to the musician intention. When it is *released*, it will join to the other agent in space and both will be aware of each other and the human musician. This awareness property influence in their movement paths and their behaviour as a *swarm*.

The user can repeat this process as much as he or she can and have several agents in space. Additionally, the agents can be *"caught"* while they are flying around with the *air tapping* gesture. The user just have to localize where they are and *tap* on them remotely to bring one back to the *spatial positioner*. If that happens, the user takes back control and will be able to change the music line and the sound synthesis parameters, then release the agent again.

During all this session, the musician is experiencing 3D sound produced by the agents as well as visualizing them as 3D objects in the physical space. The user can interact constantly with them by walking around the room to feel the effects of their movements, changing the tempo in real-time, and *catching* or *releasing* them indefinitely, until the performance is manually stopped entirely.

This chapter describes how the system explained in this section is brought to reality.

4.1.2 Intended Users

This system is intended for musicians with several levels of music improvisation skills that are interested in *experimental music* through artificial intelligence.

The goals for these users can vary depending of their interest. It can be used as an inspirational tool for assisting in the music compositional process, self-entertainment, or contemporary live music shows by adding capturing strategies to display the performance to an audience.

4.1.3 Scope

The system is limited to the interaction constraints described in the previous section 4.1.1. The design approaches are framed in the literature exploration presented in Chapter 2, and the implementation is determined by the available resources listed in section 4.3.1. Furthermore, there is no a high complexity in the chosen algorithms due to time restrictions, and neither an extensive review of potential solutions or possible combinations.

4.1.4 Functionalities

This section lists functional and non-functional requirements according to the overall description and scope. As a general requirement, the system should work under one only piece of software that can be easily connected with subsystems for *audio*, *visual*, and *motion tracking*.

Musical Input

- The musical material should be given by a traditional MIDI keyboard as a piano.
- The sound for the musical notes should be generated through a digital synthesizer under the MIDI standard.
- The MIDI keyboard should provide at least: 8 knobs for sound synthesis control, 8 buttons for general control and looping, and one control for changing octaves.

Multi-track Looper

- The general controls for the looper should consider: Start, Stop, and Clear all (remove all tracks and restart the system).
- The controls *per track* should have: Start record, end record, play, stop, and clear. The last one (clear), stops the loop and removes the training material used for the music generation algorithm.
- It should record only MIDI messages.
- The recording material should be composed at least of musical notes (note-on, note-off messages).
- A metronome should be used as a guide for a timed recording. It would be activated during the whole performance.
- The user should be able to change the musical tempo from the metronome by using a tap counter.

Sound Generation

- The system should support the integration of a VST plugin as the sound generator (a digital synthesizer).

- The note MIDI messages should trigger sounds only from the digital synthesizer.
- The digital synthesizer should have at least: a low pass filter, a reverb module, a delay module, an amplitude envelope, and presets for fast configuration of variety of sounds.
- Parameters from the previous effects should be mapped to the knobs of the MIDI controller.
- The number of voices for the synthesizer should be adjusted according to the computational resources available.
- Every track should have a different instance of the digital synthesizer.

Physical Place

- The room should be a enclosed area that minimize reverberation.
- There should be objects like tables, chairs, or other easily-move furniture that can be used to place the *spatial positioner* if needed.
- One of these tables is close to the center to support the MIDI controller and made it reachable to the user.
- Considering the previous point, the user would be standing most of the time during the music performance.
- The dimensions should be enough for walking around without problems.

Audio Output

- The digital synthesizers are the ones that contribute to the analog audio output.
- The output signal should be spatialized through a 3D sound solution, which would render audio to a loudspeakers array placed in the room.
- The loudspeakers array should have at least 8 units distributed around a center point of the room.

Motion Tracking

- The *spatial positioner* should be a light and hand-friendly trackeable object in space, that is, its Cartesian coordinates should be known regarding the center of the room.
- The *spatial positioner* position must match the 3D sound placement and the visuals.
- The trackeable area should cover the zone intended for the music performance.

Visualization

- The sound sources (agents) should be visualized in the room and must match the audio placement.
- The agents must be virtual augmentations since physical representations could collide against the performer or themselves.

Interaction

- The user should be able to move the *spatial positioner* easily and place it anywhere in the room.
- The user should *release* or *lock* and agent, regarding the *spatial positioner*, through a *remote air tapping* gesture.
- The user should be capable of *catching* an agent in the air by using the gesture mentioned before.
- The musical material recorded in the looper could be updated every time an agent is *locked*, if needed.

Agents Behaviour

- The music generation algorithm should take the recorded material from the looper as training data, and trigger new material with a similar style.
- The movement algorithm should allow an agent to be aware of the performer and the other agents, so that it can decide the paths to travel.

The next section illustrates the system design based on the requirements presented above.

4.2 Design

This section presents the design decisions that fulfills the requirements listed previously. This set of guidelines is based on the vision of the author expressed in section 4.1.1, as well as the theoretical framework from Chapter 2.

4.2.1 Overview and Justification

The system design is summarized in section 3.2 from the previous chapter. The same overall architecture described in that section is prompted in Figure 4.1.

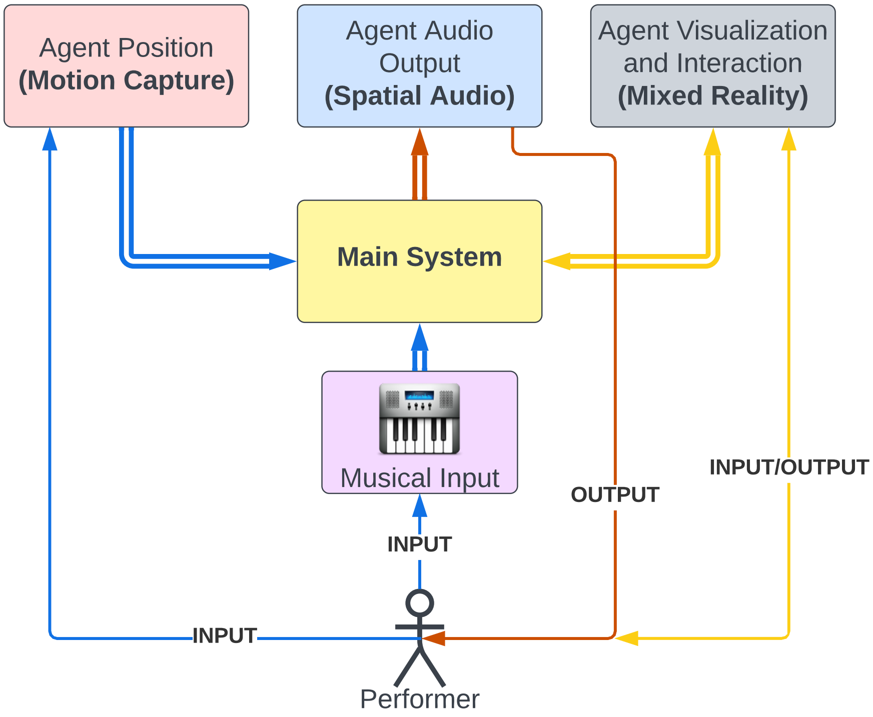


Figure 4.1: *System Design Overview.* The wider arrows represent inputs and outputs from the component to the system and the thinner ones show inputs provided by the user and outputs received from the system.

To emphasize the description provide in 3.2, the system relies on three advanced technologies for its feasibility: *motion capture*, *spatial audio*, and *mixed reality*. The main challenge to face is the conception of this solution as an effective and efficient integration of these technologies.

To achieve this goal, the system is designed by using criteria from paradigms related to *Musical Agents*, *Live Algorithms*, *Swarm Intelligence*, and *Musical XR*, as described in Chapter 2. These fields support the vision proposed since this system intends to extend music-making capabilities to manipulate the musical material produced in an improvisation across the space, as well as deal with autonomous entities as musical partners.

The design contemplates a system able to provide the ability to find novelty in a environment that can support both, familiar elements and innovative components for music improvisation. A such, the following sections depict and justify architectures for several components that assemble the whole solution.

4.2.2 Musical Input

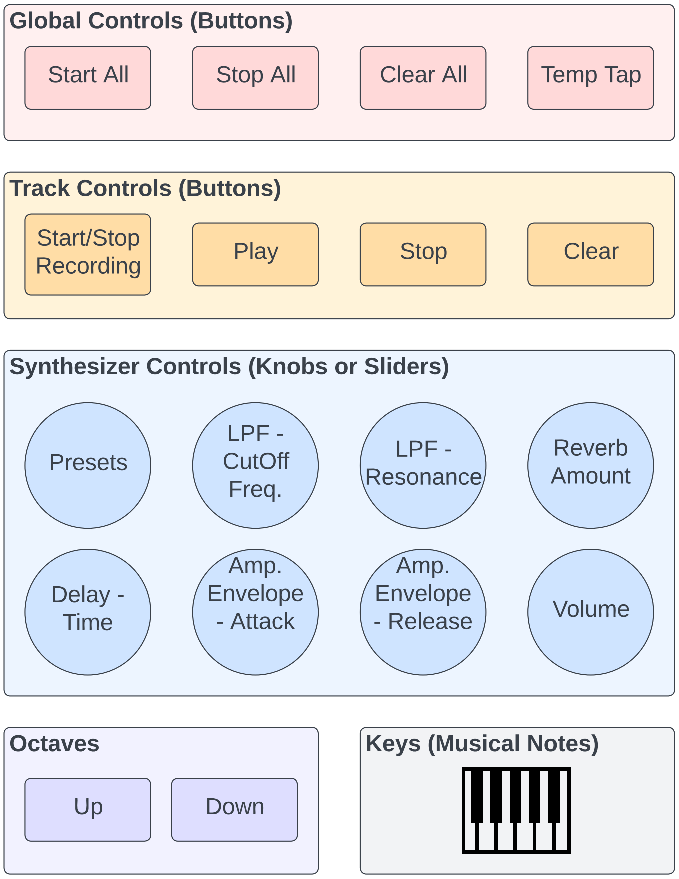


Figure 4.2: Layout for a controller that allows a user to input musical material to the system.

The user is intended to improvise music through a familiar interface. Thus, a physical interface based on a piano keyboard is suitable as standard receptor of music material. Figure 4.2 displays a layout for a controller that covers the basic controls to input musical notes, control the looper, and modify sound synthesis parameters.

This input is managed through a standard MIDI controlled whose messages are received and processed by the system.

4.2.3 Multi-track Looper

The looper designed for this application is depicted in Figure 4.3. The data recorded are MIDI messages that represent musical notes, that is, a set of (note, velocity) pairs. The output is in a similar format and feeds a sound generator for audio playback.

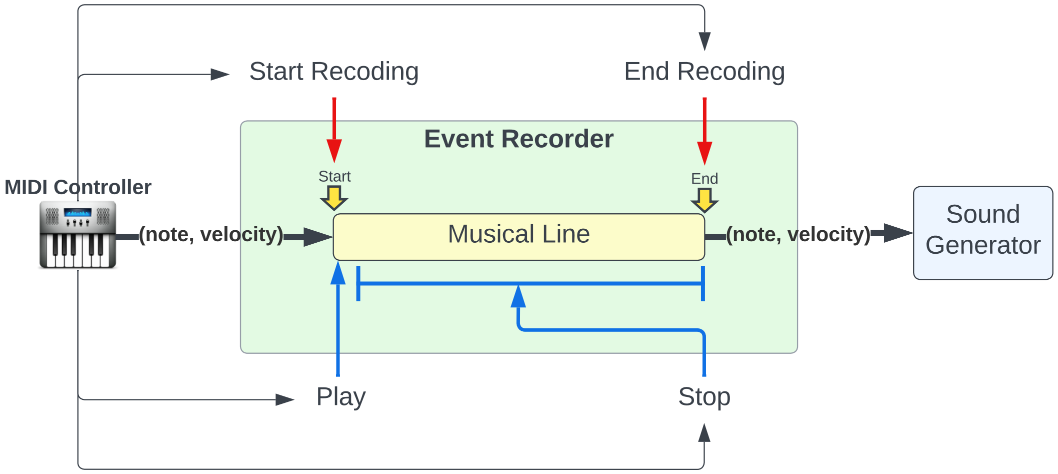


Figure 4.3: Looper operation. It allows the user to record (note, velocity) MIDI messages for endless playback.

The amount of loopers of this kind are instantiated according to the number of tracks (agents) handled by the system as shown in Figure 4.4. All of them are synchronized through a global metronome that keeps track of the time for the overall improvisation session.

The decision for this constraints has to do with the principle of *sonic organization* stated by Turchet et al. (2021) for Musical XR systems.

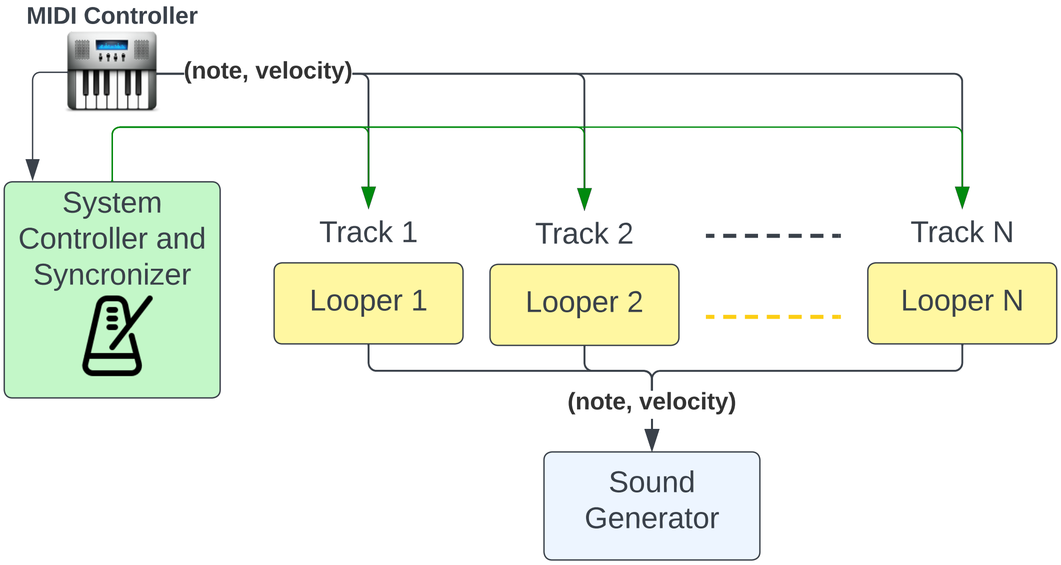


Figure 4.4: Multi-track looper configuration. There is one looper per track, that is, one looper per agent. it is synchronized by a global metronome controlled by the system.

4.2.4 Sound Generation

The audio output is produced by a digital synthesizer whose structure is shown in Figure 4.5. This module is capable of interpret (note, velocity) messages into an audio signal that can be customized in terms of synthesis parameters. This customization can be achieved through the use of a MIDI controller over several effects in the signal chain, as depicted in the image.

Every track (agent) has its own sound generator since the system is intended to provide a multi-track solution. With this configuration, different sounds can be played on every track. Figure 4.6 illustrates how any (note, velocity) source can use a sound generator per track, then, every output is spatialized for the final playback in a 3D sound setting (loudspeakers array).

This approach does not consider actual analog instruments because it increases the complexity for processing musical material based on events. The system is modeled as a *Live Algorithm* as explained later in section 4.2.9, and strategies under these type of architectures are mostly focused on discrete events (Blackwell et al., 2012).

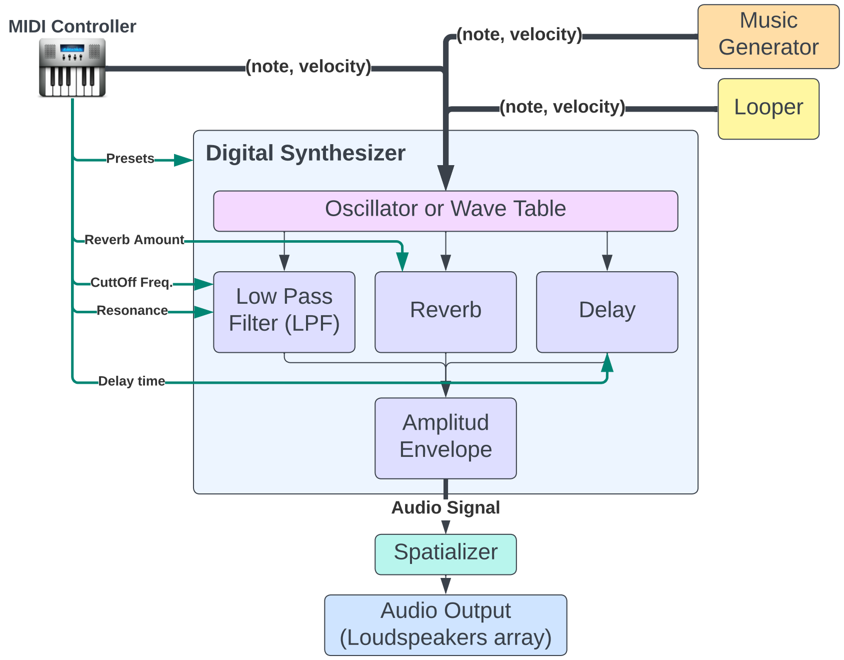


Figure 4.5: Sound Generator and synthesis modules. A digital synthesizer produces the audio output for (note, velocity) messages from several sources. It provides flexibility to change the sound properties through sound synthesis parameters.

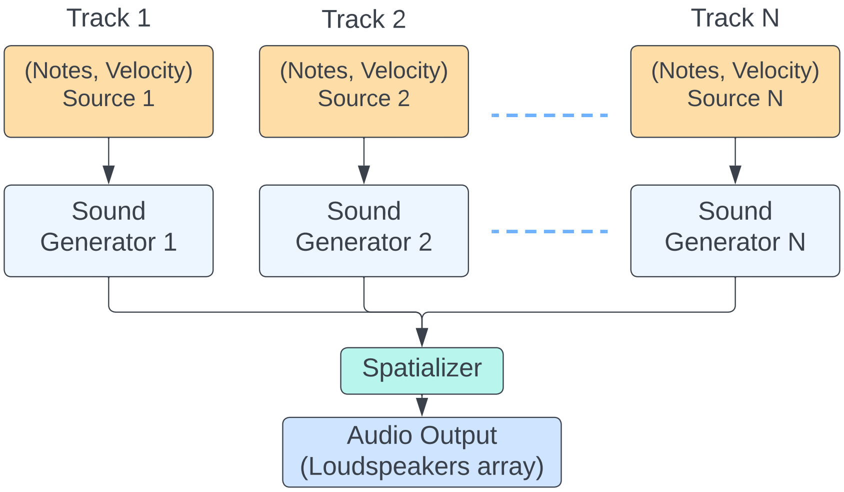

Track 1
Track 2
Track N
(Notes, Velocity) Source 1
(Notes, Velocity) Source 2
(Notes, Velocity) Source N
Sound Generator 1
Sound Generator 2
Sound Generator N
Spatializer
Audio Output (Loudspeakers array)

Figure 4.6: Sound Generator instances for every track (agent) in the system. The output is spatialized and then send to a 3D audio system.

4.2.5 Spatial Audio

The proprioceptive and auditory modalities are considered for experiencing 3D sound in this solution. As depicted in Figure 4.7, the sound generators send their signals to a *spatializer* module that place the sound sources in space through an ambisonic strategy. The encoder receives control signals to move them across the aural landscape, and then the result is decoded to a set of loudspeakers array placed in the room.

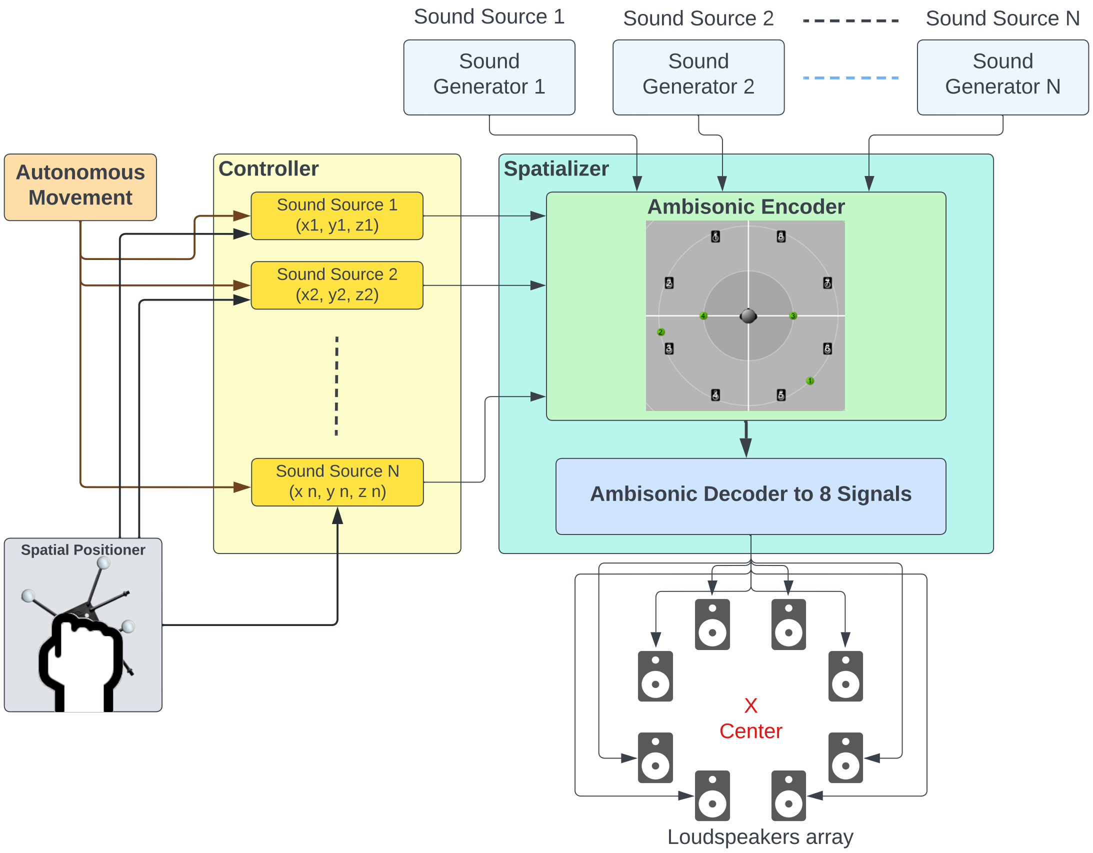


Figure 4.7: The spatializer receives all the *sound sources* (agents) and their corresponding control signals to arrange then in a 3D space through ambisonics. The agents can be manipulated either, by the *Spatial Positioner* or *autonomous movement.*

The sound sources (agents) can be moved by the *spatial positioner* or an *autonomous movement* according to their state. This data is informed to a controller that set the right coordinates to place them in the sound environment.

4.2.6 Motion Tracking

The only physical object tracked in terms of motion is the *spatial positioner*. For this purpose, it is needed a motion capture system that allows a precise position detection in the room where the performance takes place. Figure 4.8 illustrate the performance area as a green circle where the *spatial positioner* is detected. Its location determines the place from where an attached sound source (agent) is heard through the spatial audio system. An optical motion capture is recommended for this task because of its precision and previous use in spatial audio applications (Müller et al., 2014) (Grani et al., 2015).

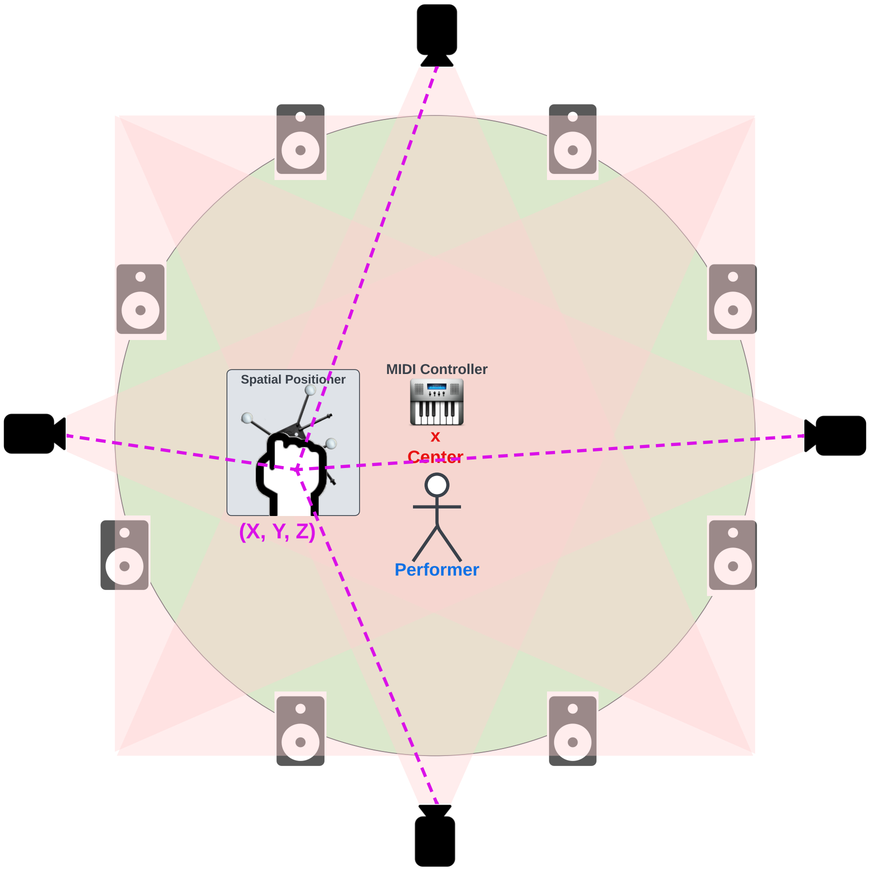


Figure 4.8: The *Spatial Positioner* being tracked in the performance area. Its position is estimated by set of motion sensors spread throughout the room.

4.2.7 Visualization

Visual is an additional modality to reinforce the human sensory experience in this application. The sound sources (agents) should be able to be identified in terms of video images for a precise location in space. This modality is important for this particular type of system because visual cues link cause and effect, leading convincing and effective performances (O'Modhrain, 2011), and they are the embodiment of the overall algorithm in the system (Blackwell, 2007).

A logical solution would be the inclusion of actual physical objects as representations for the agents (e.g. a swarm of nano drones), nevertheless, such solution would increase the complexity in terms or autonomous movement and interaction.

In that sense, a *mixed reality (MR)* strategy is suitable for this system, since it bridges sensory modalities and increases the affordances (Turchet et al., 2021). Figure 4.9 shows a virtual representation of a human performer using a mixed reality headset and observing the agents around the room.

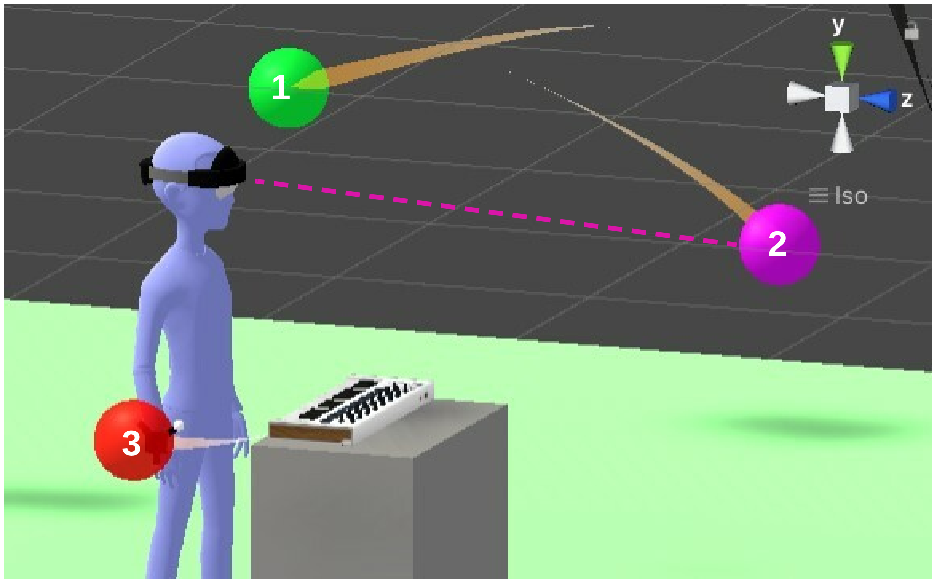


Figure 4.9: Agents visualization through a mixed reality headset. They are represented as colored spheres with numbers that identify their track number. In this image, the attention is directed to agent 2.

It was decided to represented the agents as colored *spheres* in space that left a trail as an indicator of their trajectory. This shape was selected due to its capacity to bring human attention, since it is part of a positive evolutionary mechanism associated to curved objects (Lima, 2017).

Figure 4.10 displays how an agent sphere would be attached to the *spatial positioner* when it is controlled by the user.

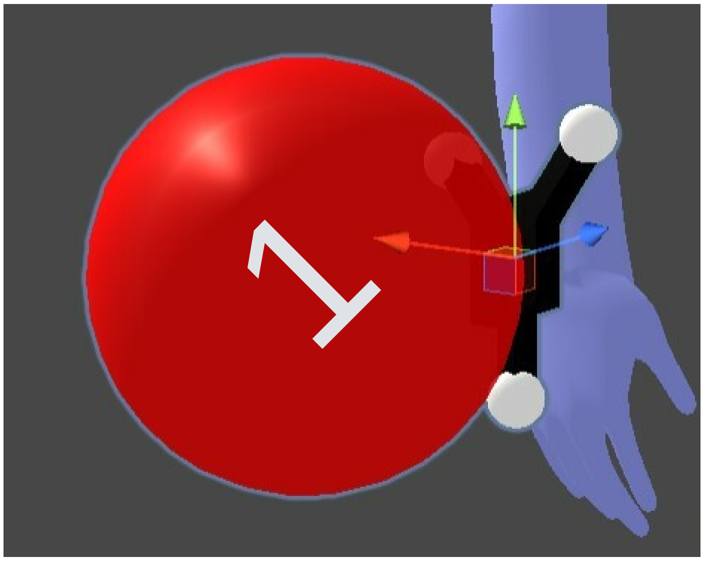


Figure 4.10: Agent attached to the *Spatial Positioner*.

The position and rotation determine the objects focused on the *field of view (fov)* of a mixed reality headset. Thus it imposes limitations for meeting human eye's capacities. That is why, under the author's criteria, mechanisms such as *directional indicators*, *graphical feedback*, and *labeling*, are important to improve a XR experience.

4.2.8 Interaction

As explained so far, the user is able to interact with a *MIDI controller* and the *spatial positioner* for sound spatialization. In the case of *"catching"* and *"releasing"* agents, it is proposed a remote *"air tapping"* gesture over the sphere visualization, since agents can be anywhere to certain distances and we want to give to users the possibility to play with them independently of where they stand in the room.

The *"air tapping"* gesture[1] is demonstrated in Figure 4.11. The user utilizes any of his or her hands and the index finger to point to the agent-as-sphere object, and *tap* over it to trigger the corresponding behaviour.

[1] https://support.microsoft.com/enus/help/12644/hololens-use-gestures

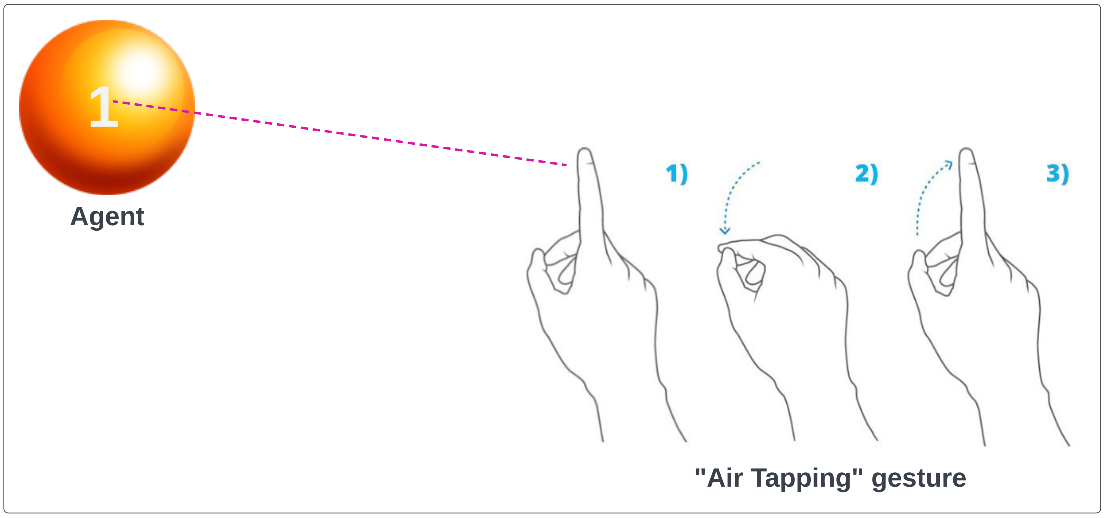


Figure 4.11: *"Air Tapping"* gesture for *"catching"* or *"releasing"* an agent-as-sphere object according to its current state.

All the interaction mechanism (*MIDI controller*, *spatial positioner*, and *"air tapping"*) are intended to be familiar and simple to provide intuition and emergency when they are combined in the system operation.

4.2.9 Autonomous Behaviour

In order to provide human-machine interaction for the proposed solution, the "machine" side needs to be defined in terms of autonomy and collaborative behaviour between the human performer and artificial entities.

For this thesis, a *multi-agent* solution based on a *swarm* of autonomous individuals is modeled under the concepts of *Musical Agents*, *Swarm Intelligence*, and *Live Algorithms*.

This section presents the design for an *Agent* as the elemental unit of the machine *music-making* and *spatialization* strategy, as well as the architecture for a group behaviour in terms of *music generation* and *autonomous movement*.

Agent Representation

According to the overall description and requirements established in the previous section 4.1, the *sound source* is an individual capable of changing the musical material from the performer and moving freely in space. Under this context of autonomy, it will be known as an *"Agent"*.

This *agent* obeys to a set of actions during a music performance session that changes its state according to the system operation. Its behaviour is described in the *Finite State Machine (FSM)* depicted in Figure 4.12.

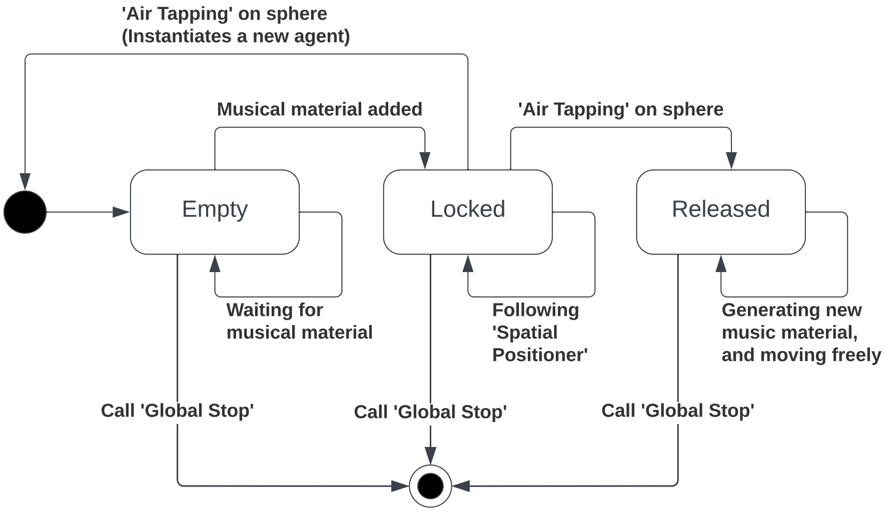


Figure 4.12: Finite State Machine (FSM) for an agent's behaviour.

Since an agent is associate with the music and sound of a track in the performance, it inherits *synchronicity* properties used for his behaviour, i.e., the global metronome rules its changes in terms of a temporal context, which is desirable for a minimal music participation (Wulfhorst et al., 2003).

Swarm Representation

The group behaviour for a set of musical agents is modeled as an *Artificial Swarm* and *Live Algorithm* under a *PQf Architecture* as shown in Figure 4.13. In this representation, there are two algorithmic approaches: *Music Generation (MG)* and *Autonomous Movement (AM)*.

As illustrated in this diagram, the *human performer* and the *released agents* listen each other in the environment (E). The analysis module (P) takes the music material from the performer to feed the *MG algorithm*, as well as the position data for *AM*. The real-time process takes this results from the (P) module to train MG models and evaluate AM formulas in the (f) module, in which new data is generated. This new material is synthesized by the (Q) module to produce spatialized sound and move the agents accordingly, then it returns to the environment (E) to start a new cycle.

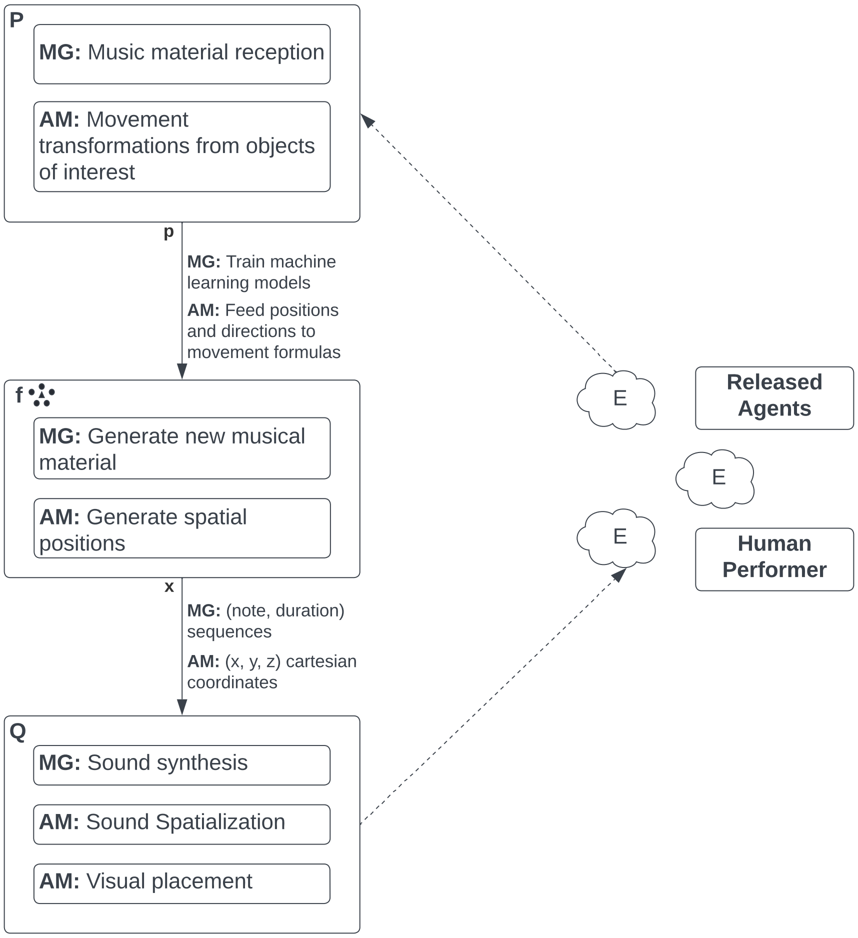

P
MG: Music material reception
AM: Movement transformations from objects of interest
p
MG: Train machine learning models
AM: Feed positions and directions to movement formulas
f
MG: Generate new musical material
AM: Generate spatial positions
x
MG: (note, duration) sequences
AM: (x, y, z) cartesian coordinates
Q
MG: Sound synthesis
AM: Sound Spatialization
AM: Visual placement
E
Released Agents
E
E
Human Performer

Figure 4.13: The system representation under the *PQf architecture* for computer music systems proposed by Blackwell (2007). *MG* stands for *Music Generation*, and *AM* for *Autonomous Movement*

Music Generation Algorithm

The intention for a music generation strategy is to produce new material based on the input from the music lines recorded in the loopers. This strategy should try to follow the performer's style for a congruent composition.

Certainly, algorithmic composition is a vast field with several techniques explored throughout the years. For this system, an algorithmic approach based on *Markov Chains* was chosen, which is a technique that generates sequences according to transition probabilities. Catak et al. (2021) recommend the use of Markov chains for music composition since it has been extensively used in music applications, and has a good quality in terms of sound frequencies and patterning to support pleasant tones.

The Markov chain's approach is taken from the strategy develop by Samuel Pearce-Davies regarding a polyphonic music generator based of multiple instances from a Markov chain's system[2] depicted in Figure 4.14.

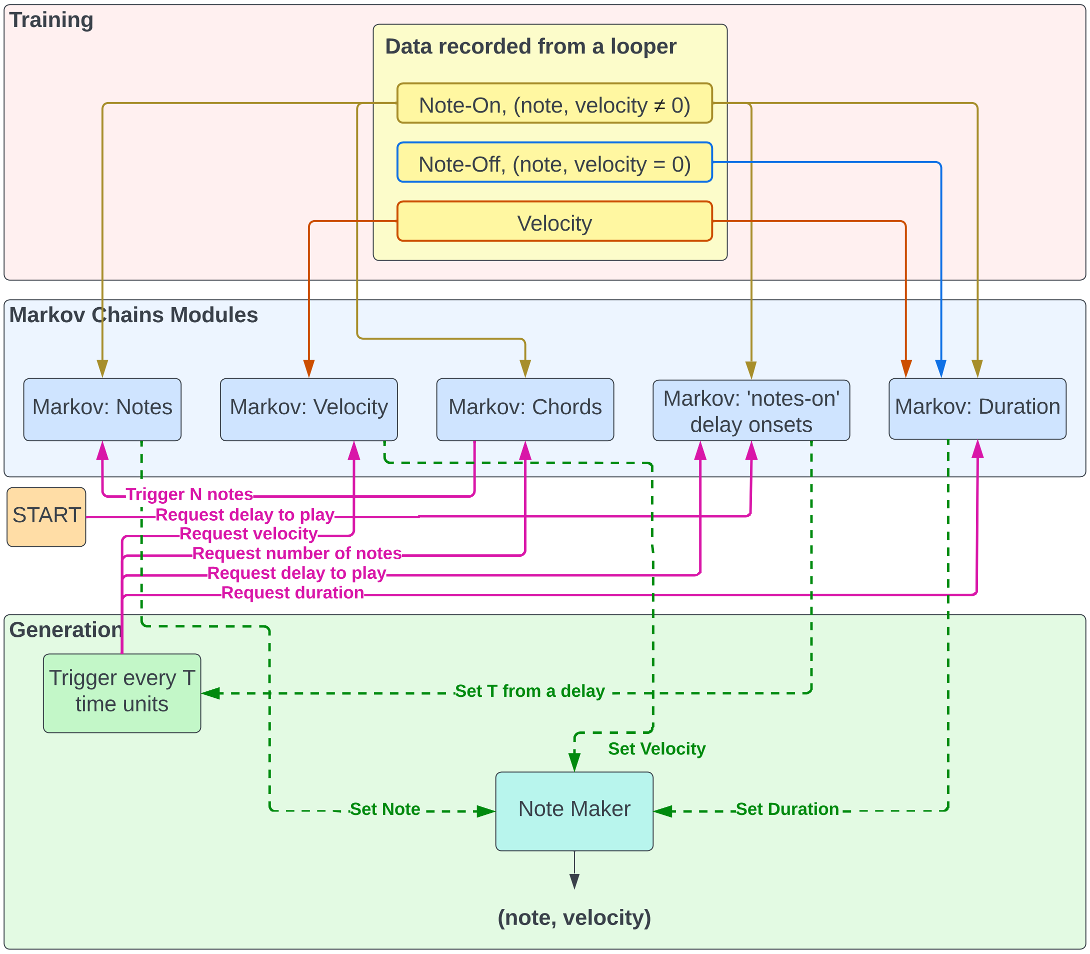


Figure 4.14: Markov chains' system designed by Samuel Pearce-Davies. It uses 5 Markov chain's modules for addressing several properties to generate material as "musical" as possible.

The system extracts from MIDI data the notes, velocities, group of notes (for chord generation), 'note-on' delay onsets, and duration. This compendium of features allows a generation of music material similar to the playing that feeds the modules since it considers melody, harmony, and rhythm as pillars for the algorithmic composition.

This solution was modified to support a synchronized triggering of new music lines according to the global system metronome. An instance of a Markov chain's system is

[2] https://spearced.com/algorithmic-process-ai/

associated with each agent as depicted in Figure 4.15. The PQf architecture, reflected in a central controller, allows the *training* and *generation* according to the overall behaviour of the system.

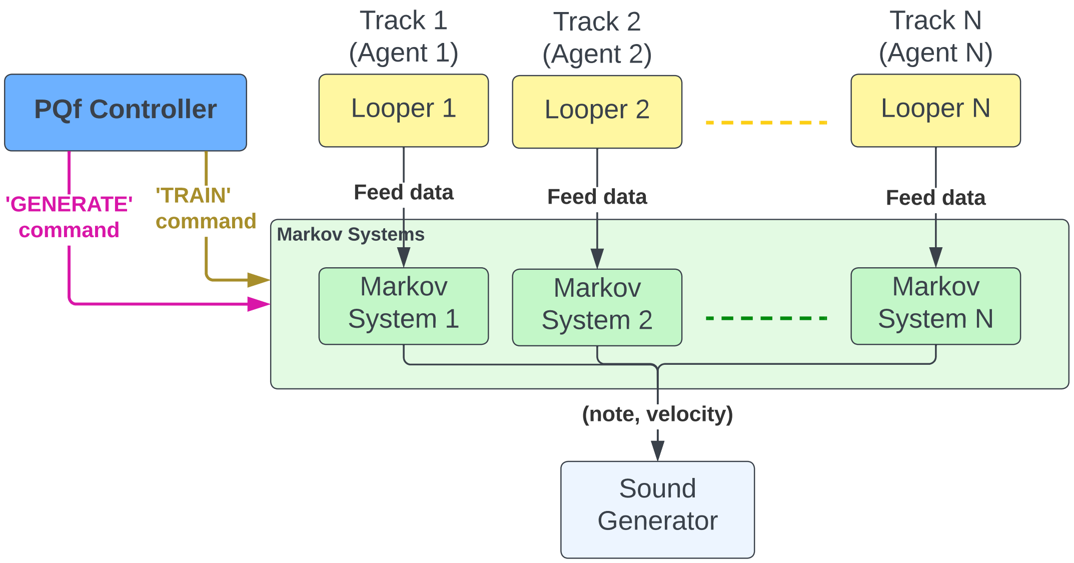


Figure 4.15: Markov chains' systems in a multi-agent approach. Several instances are associated with agents for music generation in real-time.

This design for music generation is focused on giving *sonic organization* according to the principles for Musical XR proposed by Turchet et al. (2021).

Autonomous Movement Algorithm

The agents' movement strategy is related to the *proprioceptive* modality. As a swarm, the agents should be aware of the human performer and other agents. In this context, assuming that the room center is the point $(0, 0, 0)$, the proposed algorithm considers three spatial sources to calculate the final position $\vec{P}_x$ for an individual x:

1. A circular path for a point $\vec{P}_{x:base}$ over the room center that gives a base motion.

2. For agents' co-awareness, the position $\vec{P}_{x:swarm}$ so that all of them are equally spread around the human performer position $\vec{P}_h$.

3. The human performer gaze direction given by $\mathbf{\hat{dir}}_h$.

For a smooth movement, some position calculations needs to be interpolated between points $\vec{a}$ and $\vec{b}$ during a time t. For this purpose, a *linear interpolation* model is given by (4.1).

$$\vec{P}_{lerp(a,b)} = (1-t)\vec{a} + t\vec{b} : \vec{a}, \vec{b} \in \mathbb{R}^3; t \in \mathbb{R} \tag{4.1}$$

Considering an agent x, the circular base movement for point $\vec{P}_{x:base}$ starts with spherical coordinates $(r_{x:init}, \theta_{x:init}, \phi_{x:init})$ such that r is the radius, θ is the azimuth angle, and ϕ is the elevation angle. This position is given when the user *releases* the agent x, that is, the last position where it was attached to the *spatial positioner*. The circular movement keeps the same angles $r_{x:init}$ and $\phi_{x:init}$, while the azimuth angle changes according to (4.2) with a speed S_b. Δt is the frame time of the calculation.

$$\theta_{x:base} \coloneqq \theta_{x:base} + S_b \Delta t : \theta_{x:base} = \theta_{x:init} \text{when it starts}; \theta_{x:init}, \theta_{x:base} \in \mathbb{R} \tag{4.2}$$

Thus, the circular movement for point $\vec{P}_{x:base}$, in Cartesian's coordinates, is defined by (4.3)

$$\vec{P}_{x:base} = ToCartesian(r_{x:init}, \theta_{x:base}, \phi_{x:init}) : r_{x:init}, \theta_{x:base}, \phi_{x:init} \in \mathbb{R} \tag{4.3}$$

The position $\vec{P}_{x:swarm}$ for agent's co-awareness is based on the equation for calculating the center of a group of points. In this case, this center is the performer position $\vec{P}_h$. The position of an agent is $\vec{P}_i$ in a set of N individuals, including the target x. Hence, $\vec{P}_{x:swarm}$ is computed by (4.4).

$$\vec{P}_{x:swarm} = \vec{P}_h - \sum_{i=1, i \neq x}^{N} \vec{P}_i : \vec{P}_h, \vec{P}_i \in \mathbb{R}^3; i, x, N \in \mathbb{N}^+ \tag{4.4}$$

Whereas the gaze direction $\hat{\mathbf{dir}}_h$ is a vector of 1 unit length, it will be used for the final calculation as it were a point in space 1 unit way from the center. Therefore, $\vec{P}_x$ is obtained by (4.5).

$$\vec{P}_x = \alpha(\vec{P}_{x:base} + \vec{P}_{x:swarm} + \hat{\mathbf{dir}}_h) : \vec{P}_{x:base}, \vec{P}_{x:swarm}, \hat{\mathbf{dir}}_h \in \mathbb{R}^3; \alpha \in \mathbb{R} \tag{4.5}$$

The constant α is 1/2 when the agent size is 1 and 1/3 when is greater than 1.

Additional tweaks for introducing movement variability are the following ones:

- The speed S_b is inverted ($-S_b$) when the distance between $\vec{P}_{x:swarm}$ and an agent $\vec{P_i}$ is less than m units (usually when agents are close to try to separate one from each other).

- The gaze direction $\hat{\mathbf{dir}}_h$ is inverted ($-\hat{\mathbf{dir}}_h$) or not every frame according to a random selection.

- $\vec{P}_{x:swarm}$ and $\hat{\mathbf{dir}}_h$ are interpolated in real-time according to (4.1) using a small time t_{lerp} for a smooth movement.

- S_b and t_{lerp} are arbitrary values selected by the designer to tweak the movement dynamic.

4.2.10 Integration and Communication

All the modules presented, except for *motion capture*, *spatial audio*, and *mixed reality*, can be implemented in one only self-contained system so that it can communicates with any of those three technologies through constant messages.

It implies that the system includes communication components for interfacing any device that supports the data required. Furthermore, this devices need a custom module for establishing that communication.

In this case, these subsystems can use a local network to reduce latency and packet loss, and set the connection through a wired strategy for maximizing the stability. The UDP protocol is the suggested medium for this kind of time-sensitive applications due to its minimal communication process.

4.3 Implementation

The system was implemented in the *Max/MSP/Jitter*[3] programming language (version 8.2). In the author's opinion and experience, this language provides a fast prototyping for real-time audio applications and access to an active community for developers, which was the motivation to choose it.

The three subsystems: *motion capture*, *spatial audio*, and *mixed reality*, required a separated implementation period and difference development platforms. The communication between the system and these modules was established through *OSC messages*[4] which works under the UDP protocol, except for the *spatial audio* system, that has a direct communication through the used audio platform described in 4.3.3.

This section lists the hardware and software tools that were used to implement the design proposed previously. The employment of these elements were constrained to resources availability and time.

The complementary material that includes a *video demo*, *user's sound recordings*, and the *source code* can be found in the blog post[5] on the MCT website. (https://mct-master.github.io/masters-thesis/2022/05/15/pedropl-human-machine-impro.html)

4.3.1 The MCT Portal as the Main Platform

The physical place for the system implementation was the *"MCT Portal"*, a *"laboratory for network-based musical communication"*[6] that is located in the Department of Musicology at the University of Oslo. From this laboratory, a list of the equipment with its corresponding use is listed below:

- **PC for the Main System:** The Max/MSP/Jitter implementation of the system runs in a 64-bits *Windows 10* PC with an *Intel Core i7-7700k 4.20 GHz* processor and *16 GB RAM.*

- **Audio System:** The previous PC is linked to the sound card of a *Midas M32*[7] digital console that is connected to an 8-channel loudspeakers system. The loudspeakers are evenly distributed in a circular configuration of 2 meters radius, and placed under the rooftop at 2 meters from the floor. The audio is sent directly from Max/MSP/Jitter to this system.

[3] https://cycling74.com/products/max

[4] https://ccrma.stanford.edu/groups/osc/index.html

[5] https://mct-master.github.io/masters-thesis/2022/05/15/pedropl-human-machine-impro.html

[6] https://www.hf.uio.no/imv/english/about/rooms-and-equipment/mct-portal/

[7] https://www.midasconsoles.com/product.html?modelCode=P0B3I

- **Motion Capture:** The Portal is equipped with an *OptiTrack*[8] motion capture system composed of 8 cameras that works on 120 fps and has a dedicated PC. The *spatial positioner* is represented by a *rigid body*, which is a tracking accessory from this system. (more in 4.3.4)

The performance area and setup in *The Portal* is shown in Figure 4.16. As you can notice, this is a large space where a musician can experiment the affordances provided by the system through the equipment and objects found in the room.

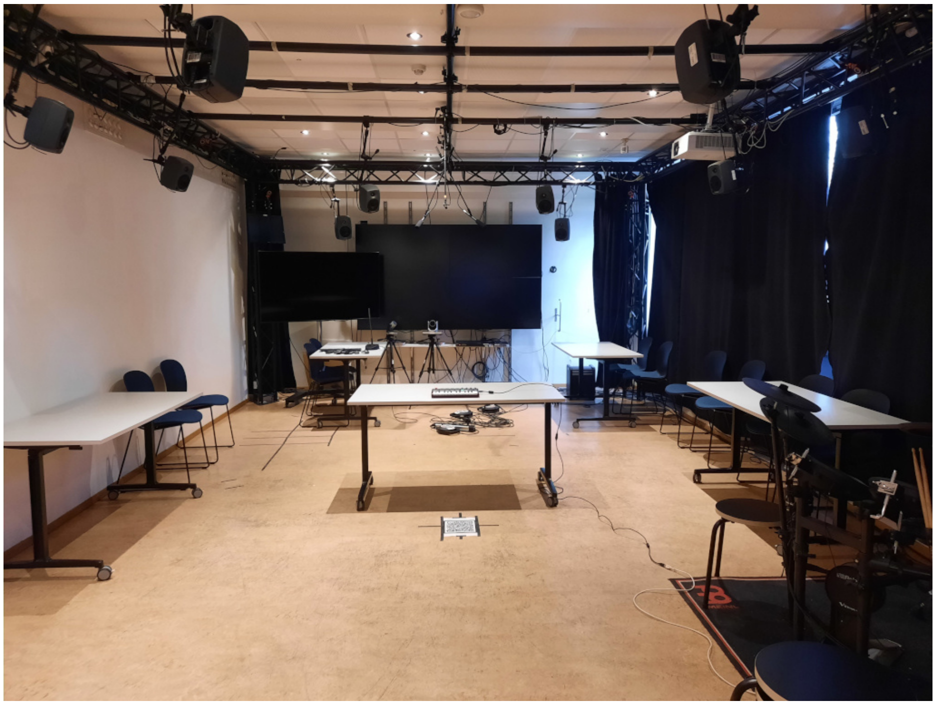


Figure 4.16: The performance area in the *MCT Portal*. The loudspeakers array for the spatial audio system is placed under the rooftop as well as the cameras for the motion capture system (no much noticeable). Moreover, it can be found in the center a table with the MIDI controller and several other furniture that can be used for placing the *spatial positioner*.

Additional equipment that is not part of the Portal is described in the corresponding sections.

4.3.2 Central Controller on Max/MSP/Jitter

The system was implemented in Max/MSP/Jitter as one only unit that runs in an independent machine. It can be considered the *"Central Controller"* for internal

[8] https://optitrack.com/

modules and the *motion capture*, *spatial audio*, and *mixed reality* subsystems.

For musical input, a MIDI controller *AKAI MPKmini II*[9] was used as the interface with the performer. This controller is depicted in Figure 4.17. The tagged buttons and knobs map the layout presented in section 4.2.2 for global control, track control, and synthesis parameters. The MIDI messages are sent to the *central controller*, which also sends back commands to light up the pads to give feedback of the metronome.

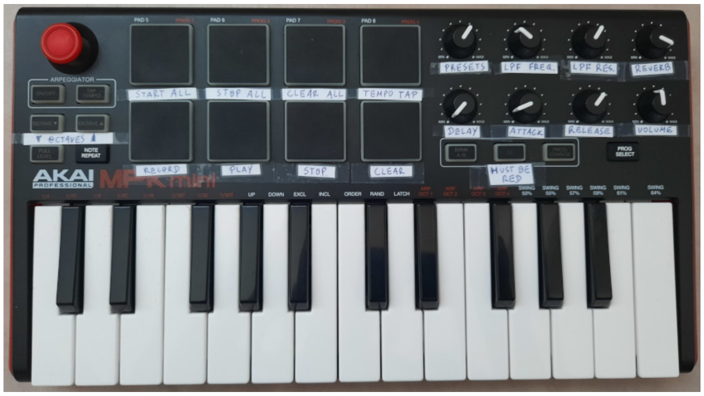


Figure 4.17: *AKAI MPKmini II* MIDI controller with controls tagged accordingly.

The global tempo is managed through the object `transport` to achieve a general synchronization of the system.

The loopers are implemented by using `mtr` objects to record events, since only (note, velocity) messages needs to be registered.

The sound synthesis is powered by the free synthesizer *Tunefish 4*[10], which is a digital sound generator in VST format that fulfills the design requirements.

For spatialization, the `spat`[11] library from *IRCAM* was integrated to apply an ambisonic approach (aep2d panning) and render analog audio to the 8-channel loudspeakers array from the *MCT Portal*.

The Markov Chains solution was taken from the source code published by Samuel Pearce-Davies[12], which uses the object `ml.markov` for Max/MSP/Jitter. Additionally, it was modified to be synchronized with the global clock of the system.

The movement behavior was implemented through JavaScript modules embedded in Max/MSP/Jitter. The chosen value for the speed of the circular base movement was $S_b = 20 degrees/s$, and for the linear interpolation for smoothing $t_{lerp} = 0.1s$. This was obtained by self-experimentation.

[9] https://www.akaipro.com/mpk-mini-mkii

[10] https://www.tunefish-synth.com/

[11] https://forum.ircam.fr/projects/detail/spat/

[12] https://spearced.com/algorithmic-process-ai/

Additionally, evaluation modules were implemented for system measurements and user recordings. This modules allowed to produce files whose analysis is presented in Chapter 5, and audio recording from each user session.

4.3.3 Spatial Audio System

The spatial audio system from the MCT Portal is described graphically in Figure 4.18. As mentioned before, it is composed by a Midas M32 console, which uses a stage box to send 8 signals for every Genelec loudspeaker in the array.

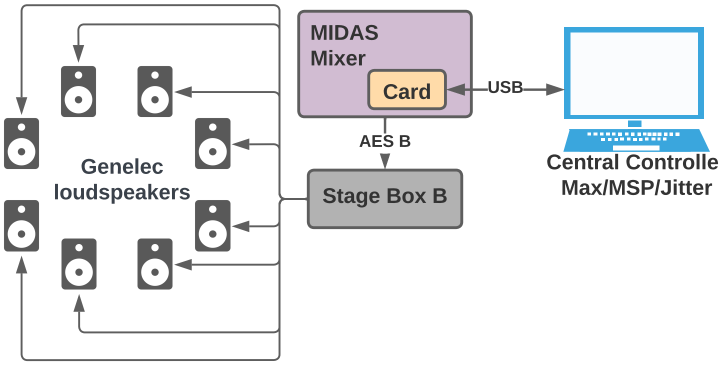


Figure 4.18: Spatial Audio System from the MCT Portal.

The system is connected through the sound card attached to the mixer, and send the decoded signal from the *sound spatializer* implemented inside the *central controller*.

4.3.4 Motion Capture as Position Sensor

The *OptiTrack*[13] motion capture system is used to track the *spatial positioner*, which physically is a *rigid body* from the OptiTrack accessories as shown in Figure 4.19.

[13] https://optitrack.com/

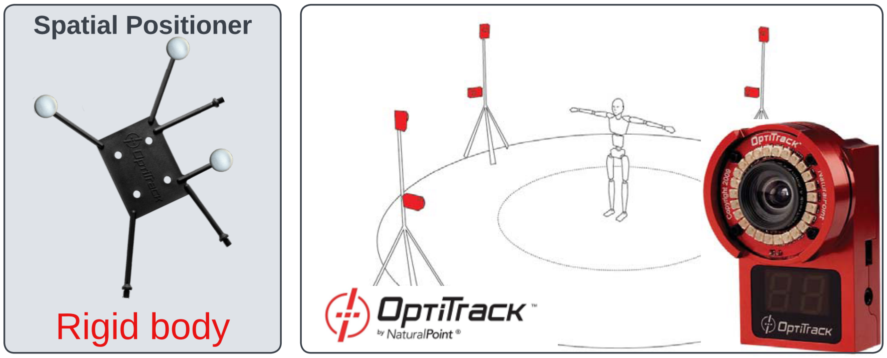


Figure 4.19: *OptiTrack* motion capture system. The image on the left is the object to be tracked. The setup and the type of cameras is shown in the right image.

For getting the position from the rigid body, a custom module for sending coordinates as OSC messages was integrated from the project *N-place*[14]. This module is developed in Max/MSP/Jitter and takes the streaming data from the *OptiTrack* software at 120 fps.

4.3.5 Mixed Reality for Visualization and Interaction

The *Microsoft HoloLens*[15] was used for agents' visualization. This mixed reality headset allows to map a physical environment with high precision and includes gestural and voice command interaction. Figure 4.20 shows an image of version 1, which was the device used for this system.

Figure 4.20: *Microsoft HoloLens* (Version 1). A high-precision mixed reality headset.

[14] N-Place Project

[15] https://www.microsoft.com/en-us/hololens

A MR application using the *Unity3D*[16] game engine was implemented to render the agents in the device. This MR application sends (commands, position, and gaze direction) and receives (agents positions, feedback) data regarding the *central controller* through OSC messages.

To calibrate the coordinate system from the HoloLens relative to the motion capture, an augmented reality module was implemented in the MR application to read a QR code placed in the center of the room. As shown in Figure 4.21, the calibration requires to look at the code to align the virtual world from the HoloLens with the physical space.

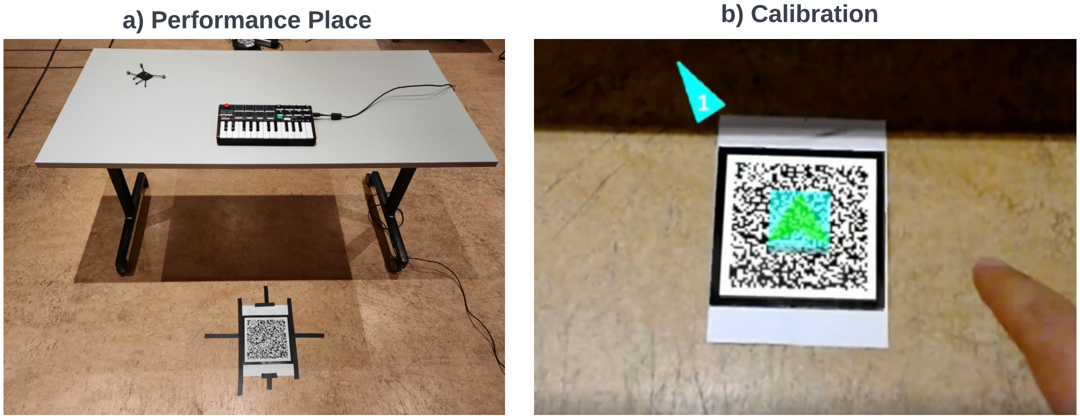


Figure 4.21: a) Performance area showing the center of the room with the QR code for calibration. b) Calibration process taking place in the HoloLens.

4.3.6 Network Communication

For communicating the motion capture PC and the HoloLens, a local network was set up through a router. The *central controller PC* and the *motion capture PC* were connect via Ethernet, while the HoloLens used the WIFI signal. The router is a 150 Mbps TP-LINK TL-WR741ND[17].

[16] https://unity.com/

[17] https://www.tp-link.com/us/home-networking/wifi-router/tl-wr741nd/

4.4 Summary

This chapter presents the design and implementation of a multimodal system for human-machine music performances. The design is guided by the concepts presented in Chapter 2 and idealized under the integration of three advanced technologies: *motion capture*, *spatial audio*, and *mixed reality*.

The core system is modeled as a *Live Algorithm* for a multi-agent solution, and meets the following behaviours stated by Blackwell (2007) based on the strategies chosen for *music generation* (Markov Chains) and *autonomous movement* (awareness formula):

- **Shadowing:** The movement algorithm includes the position and gaze direction from the performer synchronously.
- **Mirroring:** The Markov chains strategy tries to reflect the performer's style.
- **Coupling:** The agents are aware of each other by considering their positions for the real-time movement.
- **Negotiation:** The Markov chains strategy could influence the performer to change his or her way of playing to match the machine intention.

This systems was embodied in a *central controller* implemented in the *Max/MSP/Jitter* programming language. The hardware and software used for this implementation was described as well as their capabilities and limitations.

A video demonstration from the author using the system can be found in the complementary material from the blog post[18] on the MCT website.

[18] https://mct-master.github.io/masters-thesis/2022/05/15/pedropl-human-machine-impro.html

Chapter 5

Results

This chapter shows the results regarding system measurements and user testing that evaluates the design proposed in Chapter 4. The procedure was carried out by following the methodology presented in Chapter 3 under the conditions and limitations described accordingly in the next sections.

5.1 Number of Agents

The physical place and equipment described in section 4.3.1 impose limitations in terms of computational resources. The computer dedicated for the core system, and the audio platform, can handle a limited number of agents before having a significant latency and audio dropouts that affect the experience significantly.

In that sense, several trial-and-error tests were performed to find the highest number of agents possible to avoid minimally the problems mentioned above. Figure 5.1 shows a signal with a dropout that causes a perceivable audio artifact for a listener.

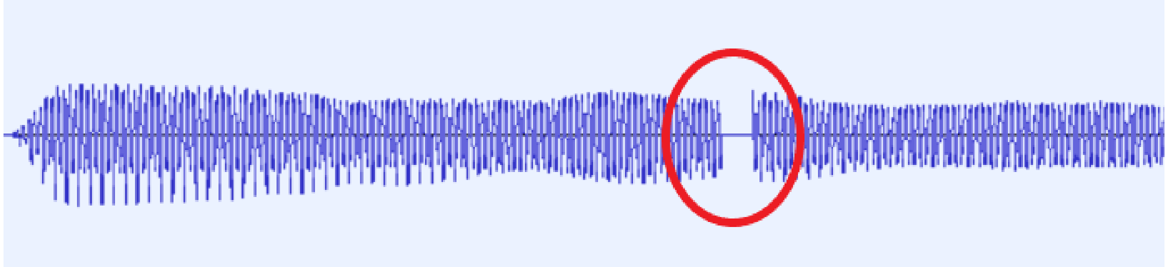


Figure 5.1: *Audio dropout.* The red circumference points out an interruption in the audio signal, which can be translated in an unpleasant cracking sound through loudspeakers.

It was found that **8 (eight)** agents can be used. The results from these tests are linked to constraints mentioned below:

- The core system is running in an independent machine and is communicated with other modules as described in section 4.3.1
- Having 8 agents requires 8 independent instances of the chosen synthesizer[1]. Every instance was configured to process up to 6 voices for polyphonic playing.
- An instance of a synthesizer includes the following modules: wave table generator, low pass filter, reverb, delay, amplitude envelope, and amplifier. Any other effect was bypassed and not processed.
- The *sample rate* for the integrated system is 441000 Hz.
- The *buffer size* for the integrated system is 1024.

The system measurements and user testing are based on this maximum number of agents as illustrated in the following sections.

[1] https://www.tunefish-synth.com/

5.2 System Measurements

5.2.1 Latency and Jitter

MIDI Controller to Sound Output Latency

The system was run with a number of agents ranging from 1 to 8, and the latency between a *key* pressed in the MIDI controller and the *sound output* was measured 30 times pear each group. This latency value is composed of two measurement points: from the *key* to the *direct sound*, and from the *direct sound* to the *listener*.

The latency from the *key* to the *direct sound* is presented in Table 5.1 per agent size. The corresponding box plots are depicted in Figure 5.2. The jitter is represented by the *standard deviation* for each case.

Agent Size	**Min.**	**Median**	**Mean**	**Max.**	**Std. Dev.**
1	34.083	40.562	42.067	53.562	6.004
2	27.188	38.427	38.802	50.667	6.445
3	25.938	36.198	37.162	50.146	6.783
4	23.312	39.625	38.760	53.000	8.036
5	26.438	37.333	38.073	50.417	6.812
6	26.625	38.938	38.724	50.792	6.731
7	22.312	35.833	36.922	51.792	7.829
8	26.458	39.073	39.810	52.292	7.049

Table 5.1: Descriptive statistic for latency from *key* to *direct sound* in milliseconds.

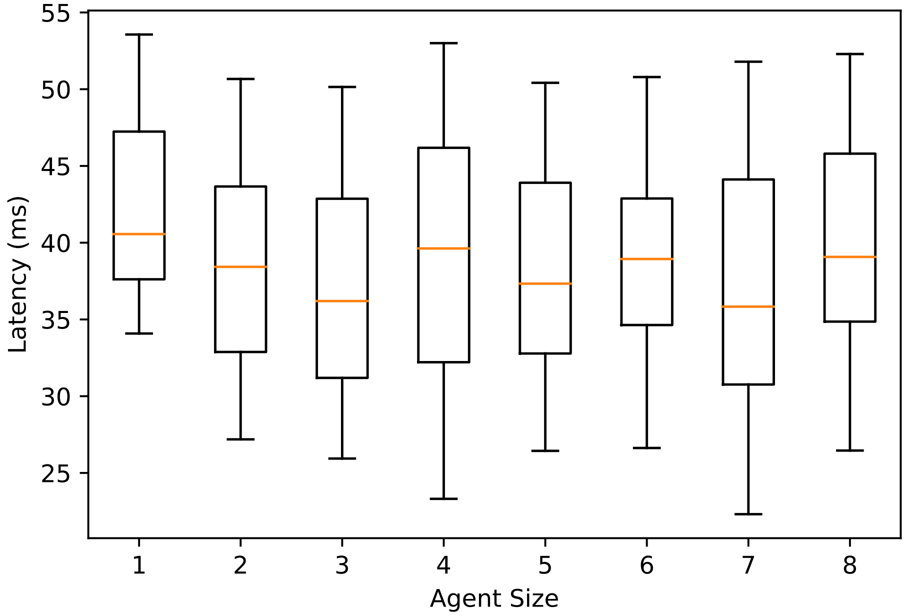


Figure 5.2: Boxplots for latency from *key* to *direct sound* in milliseconds.

Note that the central values (median and mean) and the standard deviation are similar between groups. For studying a statistically significant difference, a *One-Way Anova Test* for determining whether there are differences between the means of each group was applied, as well as a *Levene's Test* to assess the equality of variances.

The result for the Levene's Test was $P > 0.05$, which tell us that there is no significant different between variances (H_0 is accepted, H_0: *There is no difference in variance between populations*). This test enables the application of the One-Way Anova Test for means, whose result was $P > 0.05$, that is, there is no significant different between means (H_0 is accepted, H_0: *There is no difference between means of the populations*).

According to these tests, the latency and jitter from the *key* to the *direct sound* would remain constant independently of the number of agents. The total average for these values are: $latency = 38.79ms$, $jitter = 6.96ms$.

The latency from the *direct sound* to a *listener* who is assumed to be in the center of the performance area, is described in Table 5.2 and its boxplot is shown in Figure 5.3. This result was calculated from 240 samples (30 for the 8 agents) and is not organized per agent size since it represents the time that the sound travels from the loudspeakers to the listener, which is in theory the same for any case.

	Min.	**Median**	**Mean**	**Max.**	**Std. Dev.**
All Agents	5.625	5.667	5.677	5.75	0.022

Table 5.2: Descriptive statistic for latency from *direct sound* to *listener* in milliseconds.

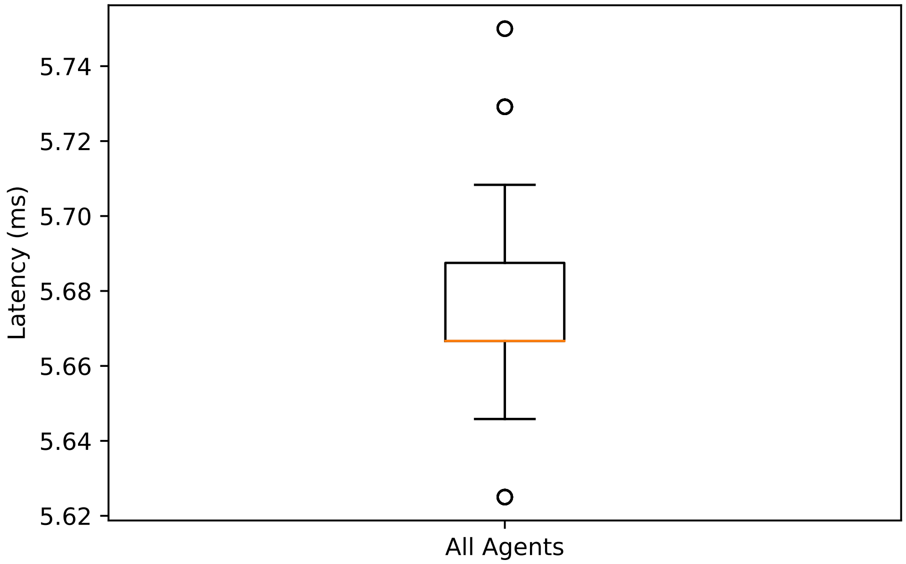


Figure 5.3: Boxplot for latency from *direct sound* to *listener* in milliseconds.

The samples were taken from a distance between the *loudspeakers array* and the *listener* of *2 m*. If we consider the speed of sound through air in a standard environment[2] (*335 m/s*), the travel time would be *5.97 ms*, which is close to the average value of *5.677 ms* found in the measurements.

The sum of these two latency values gives us the total delay from the MIDI controller to the sound output, which is approximately **44.46 ms** with a jitter of **6.96 ms**.

Spatial Audio Placement Latency

This latency refers to the delay between the *rigid body movement* and the *sound output panning* from the loudspeakers array. Its measurement was carried out in two parts: the latency between the *rigid body* and the *main system*, and the latency between the *main system* and the *spatial audio placement* in the loudspeakers. The methodology to find those values is explained in section 3.4.1.

[2] https://www.grc.nasa.gov/www/k-12/airplane/sound.html

The measurements require the use of a video camera and a screen monitor for both parts as depicted in the methodology in Figures 3.3 and 3.4. Those two elements dictates the precision of the results. The screen monitor has a delay to render the actions produced in a computer which is called *response time*. For the monitor that was used[3], the response time is *7 ms*. The video camera records in Full-HD (1920 x 1080) with a frame rate of *60 fps*.

For the first part, i.e. the latency between the *rigid body* and the *main system*, several videos were taken for each agent size. Figure 5.4 shows the setup that includes an inclined plane of approximately 20° that has a ruler attached to it.

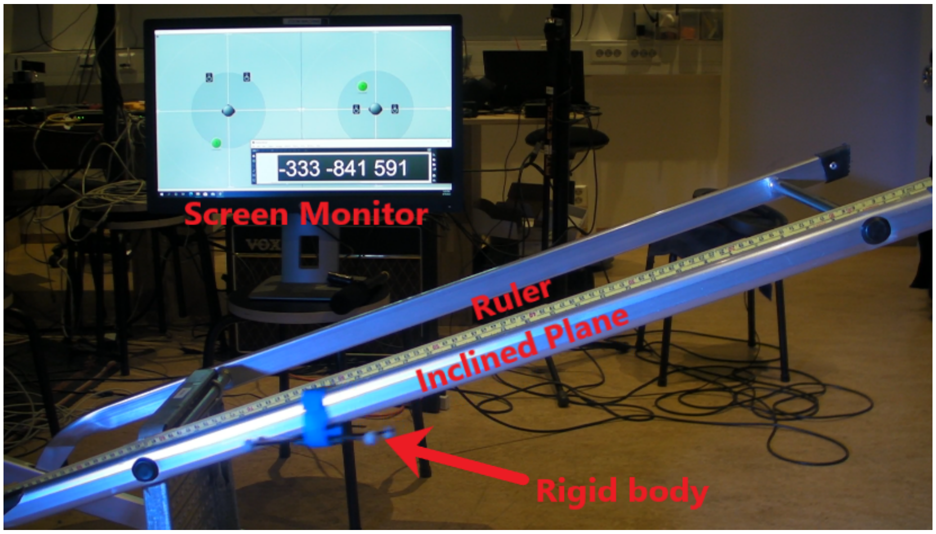


Figure 5.4: Setup for measuring the latency between the *rigid body* and the *main system*. The monitor has a response time of 7 ms and the inclined plane has an angle of around 20°.

The distance that the rigid body travels is around 80 cm. Since the rigid body speed can be different at any point of the plane when moving, different points along the plane were considered to measure the latency, however, the camera frame rate was not high enough to determine those differences as the speed changes, which means that measurements were similar with an error within 16.67 ms due to the 60 Hz camera frame rate. Thus, 6 points were taken per each agent along the plane, which were averaged to obtain the results presented in Table 5.3.

[3] https://support.hp.com/sg-en/document/c04820778

Agent Size	1	2	3	4	5	6	7	8
Mean	88.888	83.333	91.666	100.0	94.44	100.0	97.222	105.555

Table 5.3: Means in milliseconds for the latency between the *rigid body* and the *main system* per agent size.

Figure 5.5 illustrates how this latency value changes when the agent size increases. It could be speculated that the latency increases as agents are added, but due to the imprecision of the equipment, it can be taken just as an approximation for an overall latency if all values are averaged. Therefore, the latency between the *rigid body* and the *main system* is around **95.139 ms**.

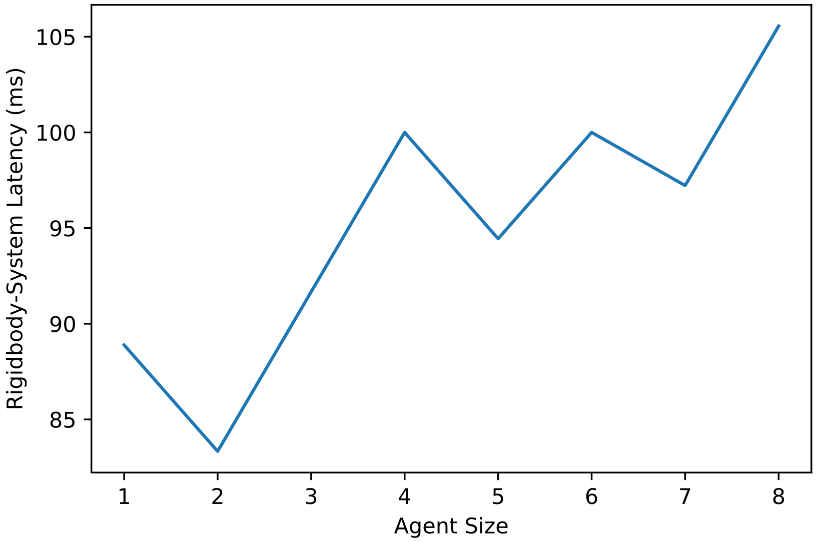
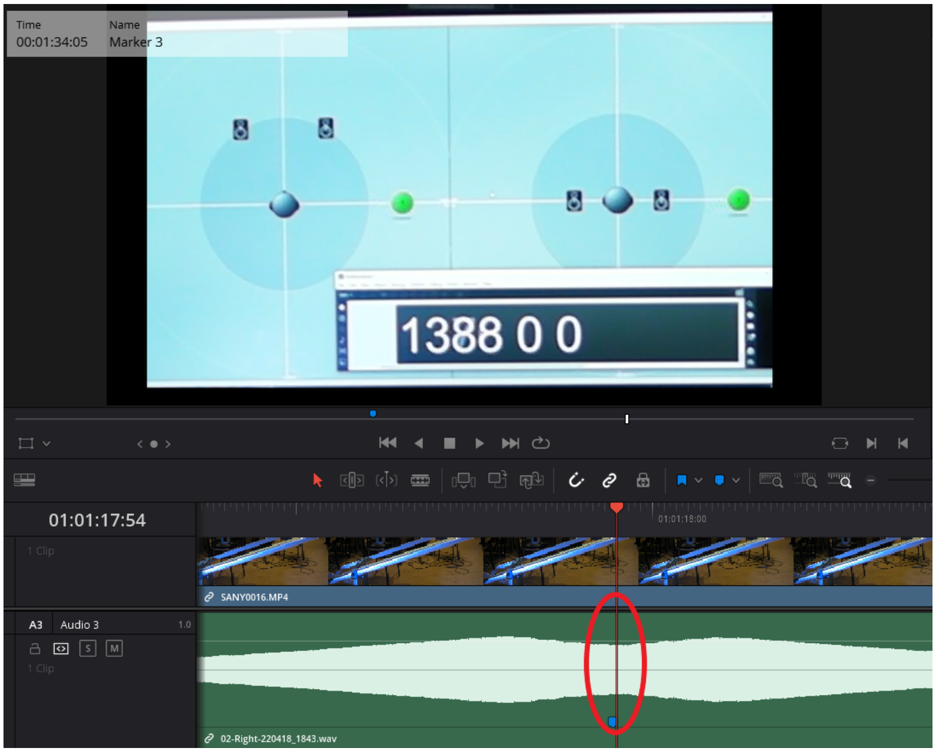


Figure 5.5: *Rigid body* to *main system* latency per each agent size in milliseconds.

The latency between the *main system* and the *spatial audio placement* in the loudspeakers is found through the inspection of the sound wave regarding a constant movement of the agent when it reaches one of the ends of the x axis, which can produce a higher signal on the left or right side of the loudspeakers array.

Figure 5.6 shows the rightmost point of an agent and its audio signal to detect this latency. For any agent size, this difference is 1 frame, which means that the latency is less than or equal to 1/60 ms (16.67 ms) since the video camera is limited to capture at 60 fps. Considering that the response time for the monitor screen is 7 ms, it is only possible to say that the latency is less than or equal to **23.67 ms** for every case. This value will be taken as the reference for this part of the latency measurement.


Time 00:01:34:05

Name Marker 3

Figure 5.6: Measurement point for the latency between the *main system* and the *spatial audio placement* in the loudspeakers. The rightmost position is identified in the audio signal to find the difference between the image and the sound panning regarding amplitude.

Table 5.4 shows the total *spatial audio placement latency* per agent size, which was obtained by joining the two values of latency presented in this section and the time that the listener hear the sound output from the loudspeakers (5.677 ms).

Agent Size	**Spatial audio placement latency (ms)**
1	118.225
2	112.67
3	121.003
4	129.337
5	123.781
6	129.337
7	126.559
8	134.892

Table 5.4: *Spatial audio placement latency* per agent size in milliseconds.

Note that the a jitter value is not considered due to the small amount of samples taken to calculate significant statistic parameters. This is because of the low equipment precision.

Sound to Visualization Latency

This latency is the time between hearing a sound source (agent) from a point in the *loudspeaker array* and visualizing it through the *mixed reality headset* (Microsoft HoloLens). This delay was measured in three stages producing the following latency results:

- **From the *main system* to the *"packet receiving module" in the mixed reality application*:** This latency represents the *network latency* over WIFI. It was measured considering the *round-trip* time for data packets to be sent per agent size. Table 5.5 shows the descriptive statistic for this parameter and Figure 5.7 the respective boxplots. These results consider 371 samples after discarding packets that were lost from an original pool size of 500 samples.

Agent Size	**Min.**	**Median**	**Mean**	**Max.**	**Std. Dev.**
1	7.0	11.0	12.197	47.0	5.561
2	6.0	12.0	13.456	57.0	7.012
3	7.0	12.0	15.358	66.0	9.927
4	7.0	12.0	15.348	105.0	10.733
5	7.0	12.0	16.652	103.0	12.190
6	7.0	12.0	17.108	164.0	16.055
7	7.0	12.0	16.585	78.0	10.480
8	8.0	15.0	21.105	114.0	14.834

Table 5.5: Descriptive statistic for the round-trip time from the *main system* to the *"packet receiving module" in the mixed reality application* in milliseconds.

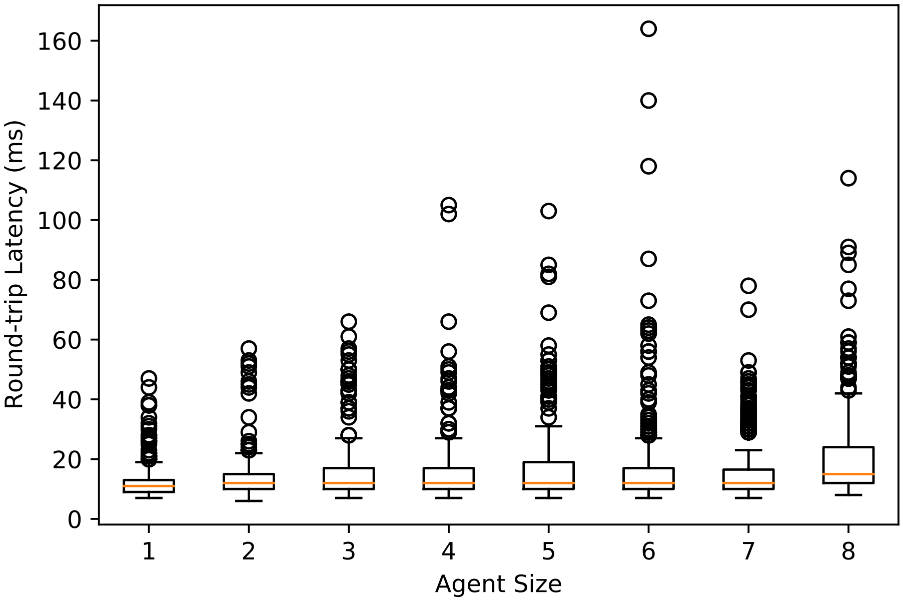


Figure 5.7: Boxplots for the round-trip time from the *main system* to the *"packet receiving module" in the mixed reality application* in milliseconds.

To determine if there is a statistically significant difference in variances and means, a *Levene's test* (difference in variances) and a *Kruskal-Wallis test* (difference in means when variances are not homogeneous) were applied. The Levene's test result was $P < 0.05$ (variances are different between groups), which influenced in the decision of using a Kruskal-Wallis test to assess the difference of means whose descriptor was $P < 0.05$ (means are different between groups).

Thus both, latency and jitter, are different per agent size as is shown in Figure 5.8 with a growth trend as agents increase.

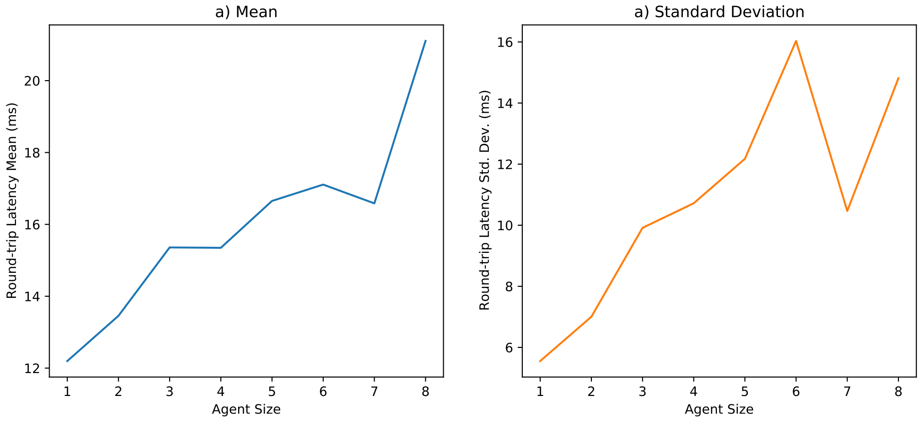


Figure 5.8: *Mean* and *Standard Deviation* for the round-trip time from the *main system* to the *"packet receiving module" in the mixed reality application* in milliseconds.

The actual latency of interest is half of the round trip, hence Table 5.6 illustrate the mean and the standard deviation as the latency and jitter per agent size in this case.

Agent Size	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Latency	6.098	6.728	7.679	7.674	8.326	8.554	8.292	10.552
Jitter	2.781	3.506	4.963	5.366	6.095	8.027	5.24	7.417

Table 5.6: Latency and jitter (in milliseconds) from the *main system* to the *"packet receiving module" in the mixed reality application* per agent size.

- **Inside the mixed reality application: from the "packet receiving module" to "position update":** This latency is the time to process the received packet and update the position for the next graphic frame inside the HoloLens in the corresponding function. Table 5.7 shows the descriptive statistic and Figure 5.9 the boxplots per agent size from 500 samples collected. The mean and the standard deviation can be taken as the latency and jitter of interest.

Agent Size	**Min.**	**Median**	**Mean**	**Max.**	**Std. Dev.**
1	0.0	9.0	8.916	18.0	4.692
2	0.0	10.0	9.050	18.0	5.090
3	0.0	10.0	9.534	23.0	4.894
4	0.0	10.5	9.648	19.0	5.099
5	0.0	10.0	10.080	20.0	4.762
6	0.0	10.0	9.634	19.0	4.834
7	0.0	9.0	9.084	19.0	4.817
8	0.0	9.0	9.070	18.0	4.769

Table 5.7: Descriptive statistics for the latency (in milliseconds) from the "packet receiving module" to "position update" inside the mixed reality application.

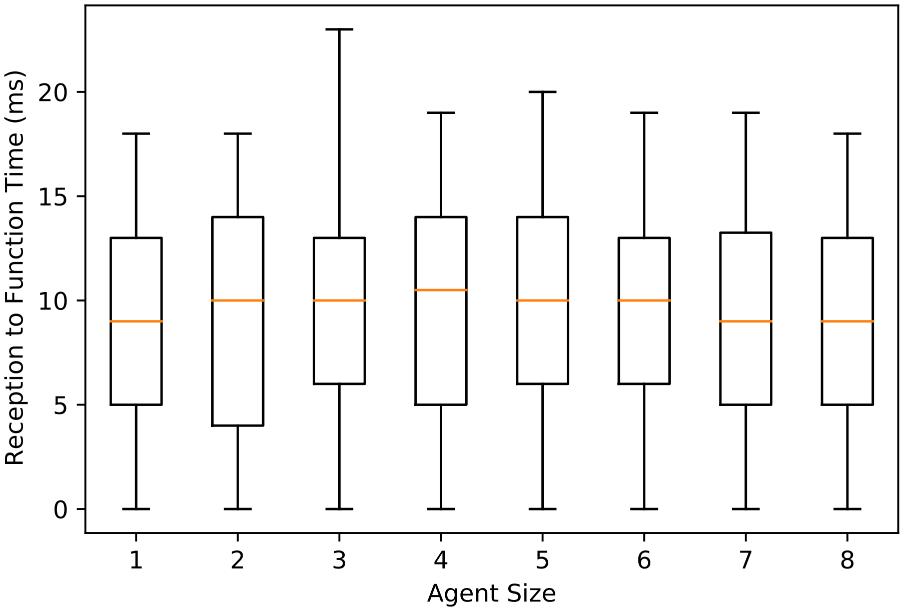


Figure 5.9: Boxplots for the latency (in milliseconds) from the "packet receiving module" to "position update" inside the mixed reality application.

As in the previous case, variances and means were tested with statistical descriptors. A *Levene's test* (difference in variances) and a *One-Way Anova test* (difference in means when variances are homogeneous) were applied. The Levene's test result was $P > 0.05$ (variances are not different between groups), leading to use a One-Way Anova test to determine the difference of means with a value of $P < 0.05$ (means are different between groups). As such, latency is different per agent size, but jitter is similar. Figure 5.10 shows the differences as agents are added. Note that in the case of the standard deviation, the y axis has a small range, which suggests the results of the variances' test.

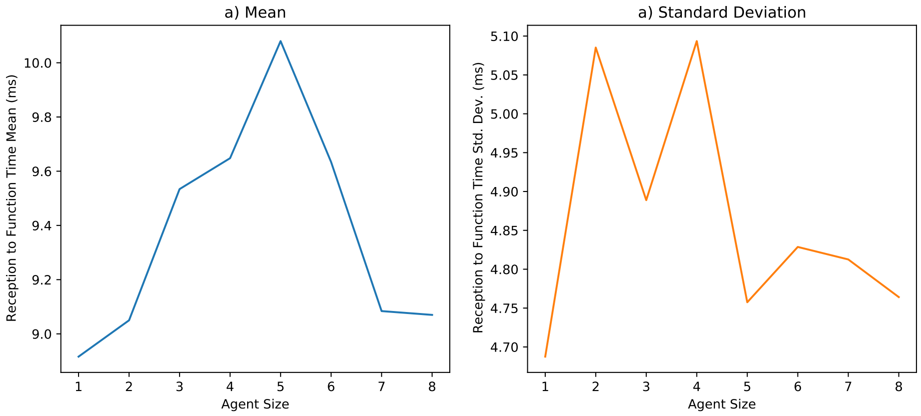


Figure 5.10: *Mean* and *Standard Deviation* for the latency (in milliseconds) from the "packet receiving module" to "position update" inside the mixed reality application.

- **Inside the mixed reality application: The "real-time application loop":** This latency is the frame time of the mixed reality application to render an image in real-time, that is, the delay between the update of an object position and when it is drawn. The results were calculated from 500 samples, and the descriptive statistic and boxplots are shown in Table 5.8 and Figure 5.11 respectively.

Agent Size	Min.	Median	Mean	Max.	Std. Dev.
1	12.0	17.0	16.568	21.0	1.152
2	13.0	17.0	16.592	20.0	1.213
3	10.0	16.0	16.526	23.0	2.444
4	9.0	16.0	16.412	26.0	2.345
5	7.0	16.0	16.292	28.0	2.031
6	8.0	17.0	16.522	25.0	2.296
7	8.0	17.0	16.584	22.0	1.598
8	8.0	16.0	16.464	21.0	1.254

Table 5.8: Descriptive statistics for the frame time (in milliseconds) in the mixed reality application per agent size.

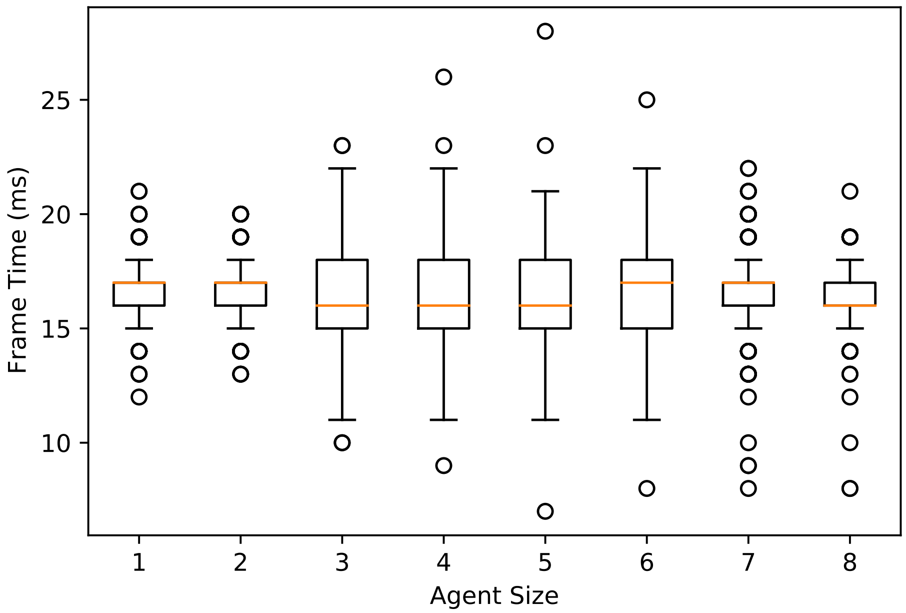


Figure 5.11: Boxplots for the frame time (in milliseconds) in the mixed reality application per agent size.

Likewise, variances and means were tested with a Levene's Test ($P < 0.05$) and a Kruskal-Wallis test ($P = 0.0366, P < 0.05$) respectively. Thus, there is a statistically significant difference for variances and means between groups. Figure 5.12 depicts those differences per agent size.

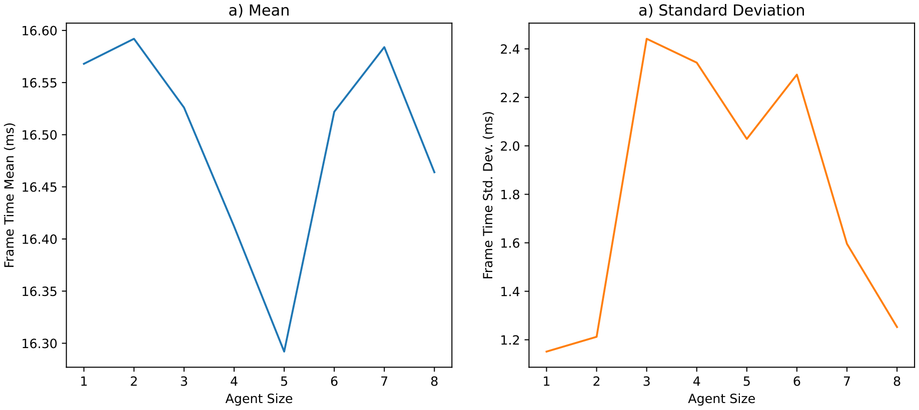


Figure 5.12: *Mean* and *Standard Deviation* for the frame time (in milliseconds) in the mixed reality application per agent size.

Considering all the data, the frame time in average is **16.49 ms** with a jitter of **1.79 ms**. Hence the mixed reality application runs approximately at 60 fps, an acceptable frame rate for a like-video-game application (Claypool & Claypool, 2007).

Let us consider that the total *sound to visualization latency* is T_{sv}, and the values for the three stages presented above are t_{sr}, t_{rp}, and t_{ft} respectively; additionally, from the previous section, let us take the latency between the *main system* and the *spatial audio placement* in the loudspeakers which is $t_{ss} = 23.67ms$ for all agents, and the travel time for the sound from the *loudspeakers* to the *listener* as $t_{air} = 5.677ms$ for all agents as well. Then, the *sound to visualization latency* T_{sv} is given by (5.1), as well as an analogy for jitter from (5.2) that takes only the three stages.

$$T_{sv} = (t_{sr} + t_{rp} + t_{ft}) - (t_{ss} + t_{air}) \tag{5.1}$$

$$J_{sv} = j_{sr} + j_{rp} + j_{ft} \tag{5.2}$$

Table 5.9 shows the results from this calculation per agent size.

Agent Size	Latency (ms)	Jitter (ms)
1	3.245	8.624
2	3.033	9.809
3	4.402	12.301
4	4.397	12.8105
5	5.361	12.888
6	5.373	15.157
7	4.623	11.655
8	6.749	13.44

Table 5.9: *Sound to visualization* latency and jitter per agent size in milliseconds.

5.2.2 CPU Usage

The CPU usage was taken every 250 milliseconds up to 50 samples for each agent size. Table 5.10 presents the descriptive statistic, and boxplots are illustrated in Figure 5.13.

Agent Size	Min.	Median	Mean	Max.	Std. Dev.
1	14.0	16.0	16.939	22.0	1.547
2	11.0	16.0	16.612	24.0	2.767
3	13.0	24.0	23.327	31.0	3.913
4	15.0	28.0	27.551	42.0	7.597
5	14.0	25.0	26.857	50.0	7.608
6	20.0	29.0	30.143	53.0	6.958
7	15.0	27.0	27.612	50.0	7.751
8	17.0	25.0	29.041	69.0	11.621

Table 5.10: Descriptive statistics for *CPU Usage percentage* per agent size.

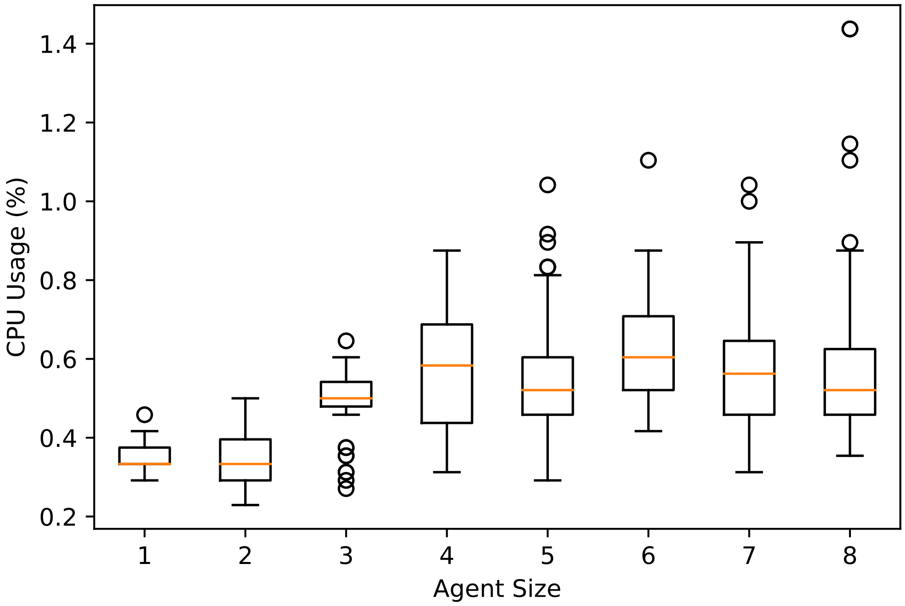


Figure 5.13: Boxplots for *CPU Usage percentage* per agent size.

This percentage raises in terms of mean and standard deviation as shown in Figure 5.14. Moreover, this results can be corroborated in how the CPU usage develops along time as depicted in Figure 5.15. Considering that the latency from *MIDI Controller to Sound Output* presented in the previous section is similar for all agents, it make sense that the CPU works more when sound processing instances are increased.

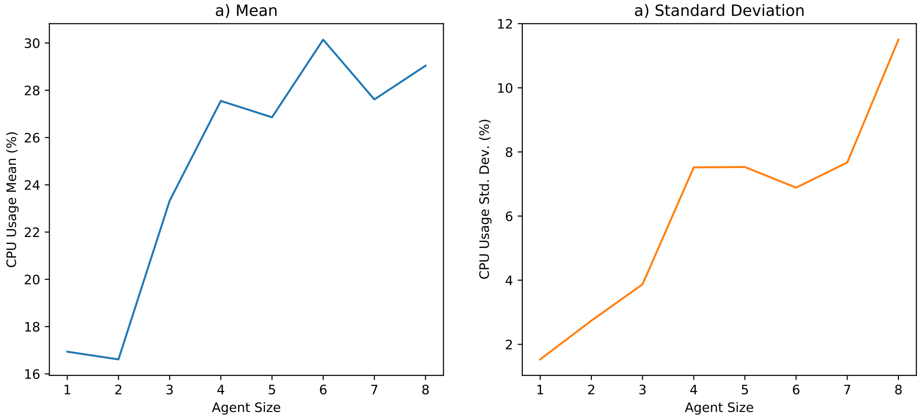


Figure 5.14: *Mean* and *Standard Deviation* for *CPU Usage percentage* per agent size.

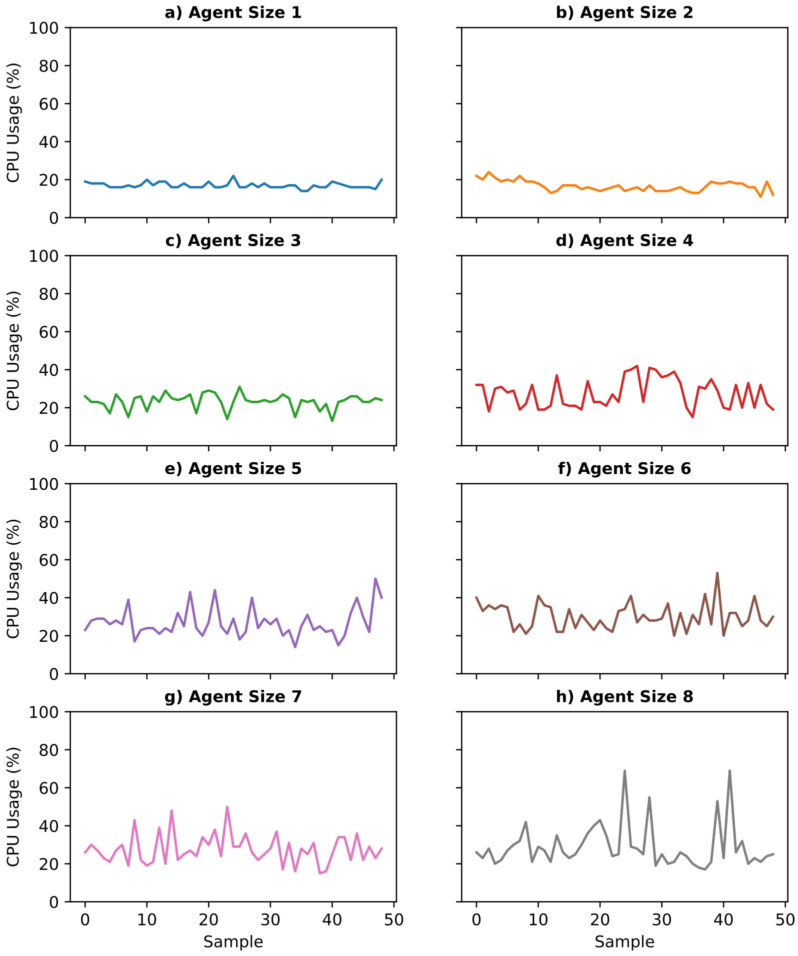


Figure 5.15: *CPU Usage* per agent size along time. The period of every chart is represented by the samples taken from the system.

5.2.3 Packet Loss

The packet loss was measured for the motion capture (Ethernet connection) and the mixed reality device (WIFI connections). Both are communicated to the main system through a local network.

The motion capture did not show any loss in a test performed during one hour, which confirms the stability of a direct connection through an Ethernet interface for a sending frequency of 120 Hz.

For the mixed reality device, packets were sent during one minute (30 ms period, 33.33 Hz). This action was repeated 30 times per each agent size. The packet loss was calculated on every repetition. Table 5.11 shows the results of these experiments and Figure 5.16 the respective boxplots.

Agent Size	Min.	Median	Mean	Max.	Std. Dev.
0	16.60	19.263	19.077	21.250	1.101
1	18.25	20.300	20.798	25.150	1.955
2	18.25	20.100	20.331	23.800	1.409
3	18.25	20.230	20.708	24.700	1.616
4	20.05	21.650	21.720	24.550	1.084
5	21.85	24.300	24.144	26.200	1.131
6	21.20	23.525	23.562	26.313	1.148
7	22.00	23.850	23.890	25.900	0.982

Table 5.11: Descriptive statistics for *packet loss percentage* regarding the communication to the mixed reality device per agent size.

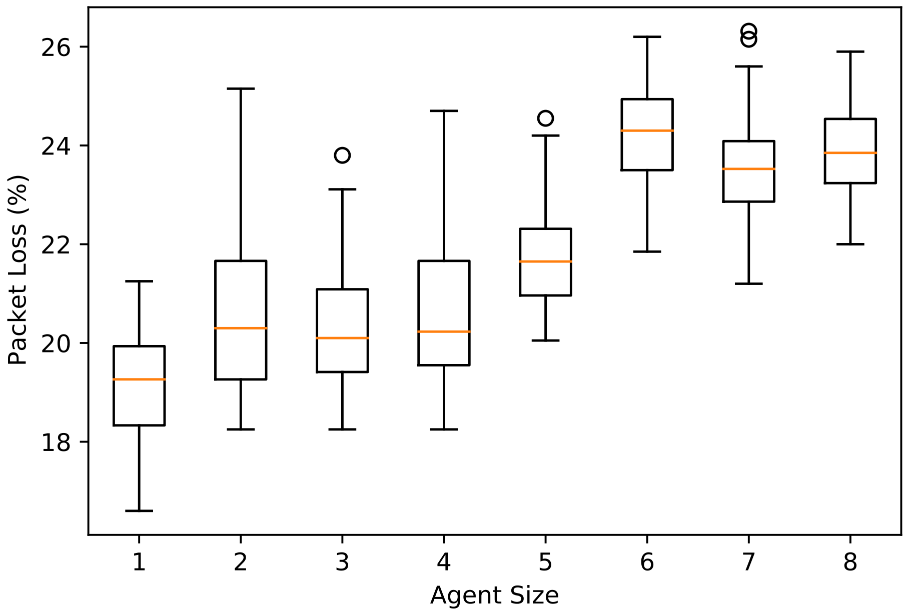


Figure 5.16: Boxplots for *packet loss percentage* regarding the communication to the mixed reality device per agent size.

It is noticeable that the packet loss percentage increases as agents are added as visualized in Figure 5.17. It can be contrasted with the results for the round-trip time from the *main system* to the *"packet receiving module" in the mixed reality application* shown in Figure 5.8 from a previous section, which basically is the *network latency*. Both have an impact when the number of agents is raised.

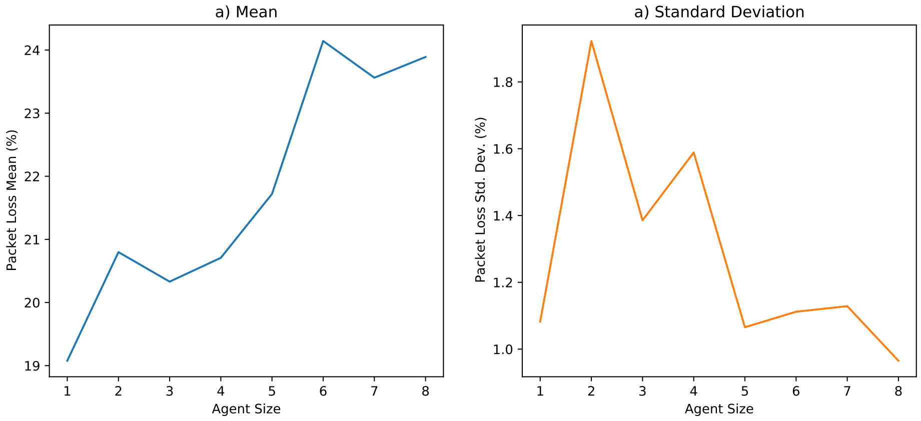


Figure 5.17: *Mean* and *Standard Deviation* for *packet loss percentage* regarding the communication to the mixed reality device per agent size.

5.2.4 Autonomous Algorithms Measurements

Music Generation Algorithm

A second-order *Markov Chains* module was evaluated to determine the impact in terms of computational time for the system. Two cases that were assessed are presented below.

- **Build time:** As described in the methodology, it considers 30 build cycles with training samples from 100 to 3000 with steps of 100. The training samples were random numbers between 1 and 120. The building time is measured 30 times for every training size and per each agent size, that is, up to 900 measurements per each agent size.

  The resulting boxplots from the descriptive statistic are depicted in Figure 5.18.

  These results suggest that the build time increases linearly as the training size grows, describing a possible function $y = ax+b$. As such, a linear regression with an estimation of goodness of 0.972 was calculated ($a = 0.0296$, $b = 0.002305$, $StdError = 6.575e-06$, $P < 0.05$). This linear model and the data related are shown in Figure 5.19.

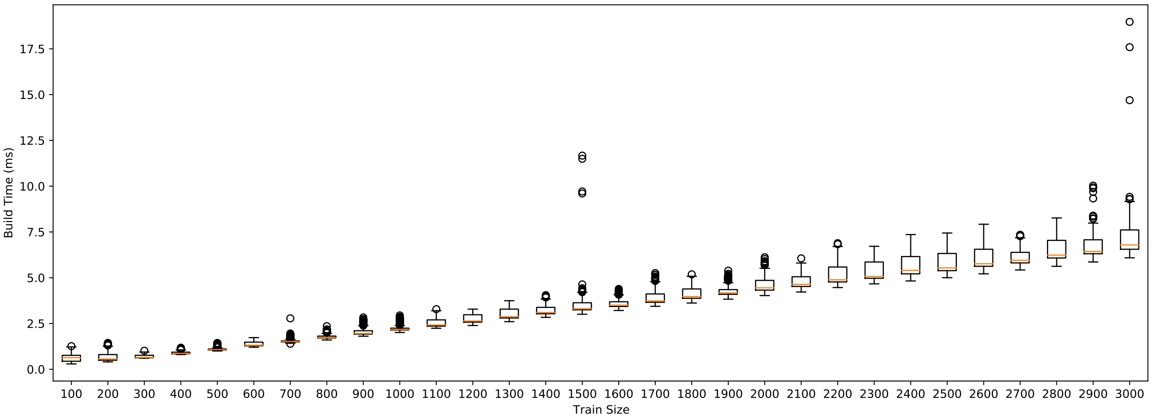


Figure 5.18: Boxplots for the *build time* in milliseconds regarding a *second-order Markov Chains module* per training size.

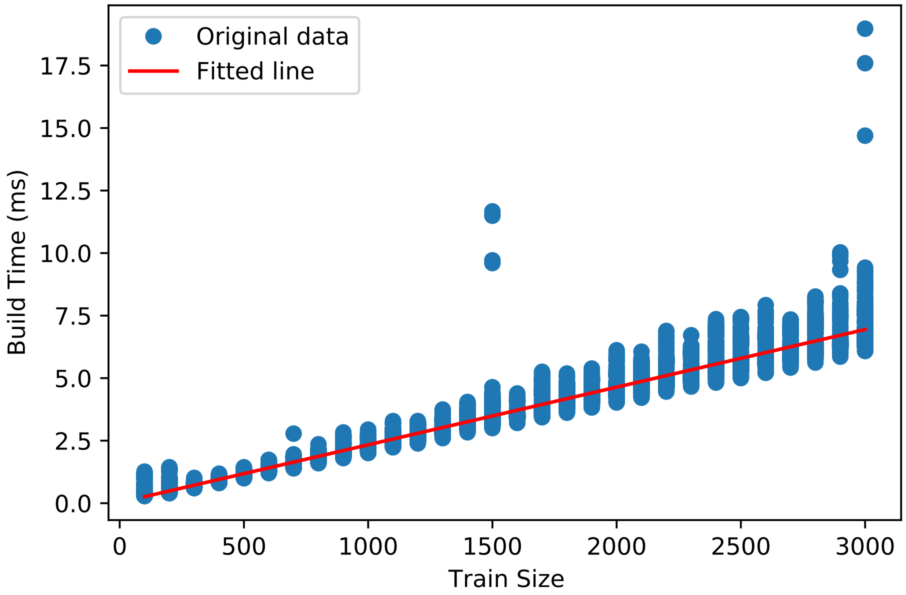


Figure 5.19: Scatter plot and linear model for the *build time* in milliseconds regarding a *second-order Markov Chains module* per training size.

- **Generation time:** This is the time that the system takes to request and generate an element from the model. For this test, 250 samples were collected per agent size. Table 5.12 shows the descriptive statistic and Figure 5.20 the corresponding boxplots for each case.

Agent Size	Min.	Median	Mean	Max.	Std. Dev.
1	0.011	0.036	0.037	0.080	0.005
2	0.013	0.038	0.035	0.047	0.007
3	0.007	0.028	0.028	0.069	0.005
4	0.008	0.022	0.022	0.037	0.005
5	0.007	0.023	0.025	0.113	0.009
6	0.007	0.021	0.023	0.052	0.007
7	0.003	0.018	0.020	0.043	0.007
8	0.005	0.016	0.018	0.044	0.007

Table 5.12: Descriptive statistics for the *generation time* in milliseconds regarding a *second-order Markov Chains module* per agent size.

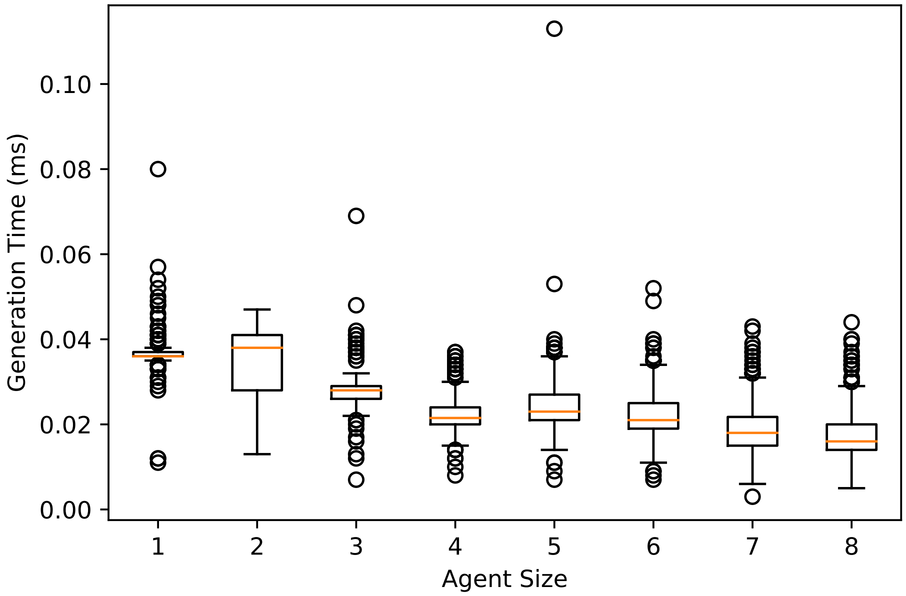


Figure 5.20: Boxplots for the *generation time* in milliseconds regarding a *second-order Markov Chains module* per agent size.

There is no a significant observable pattern for this parameter as is noticed in Figure 5.21. However, it is important to remark that these values are considerably below 0.12 ms, which suggests that the generation time is a negligible parameter for any case.

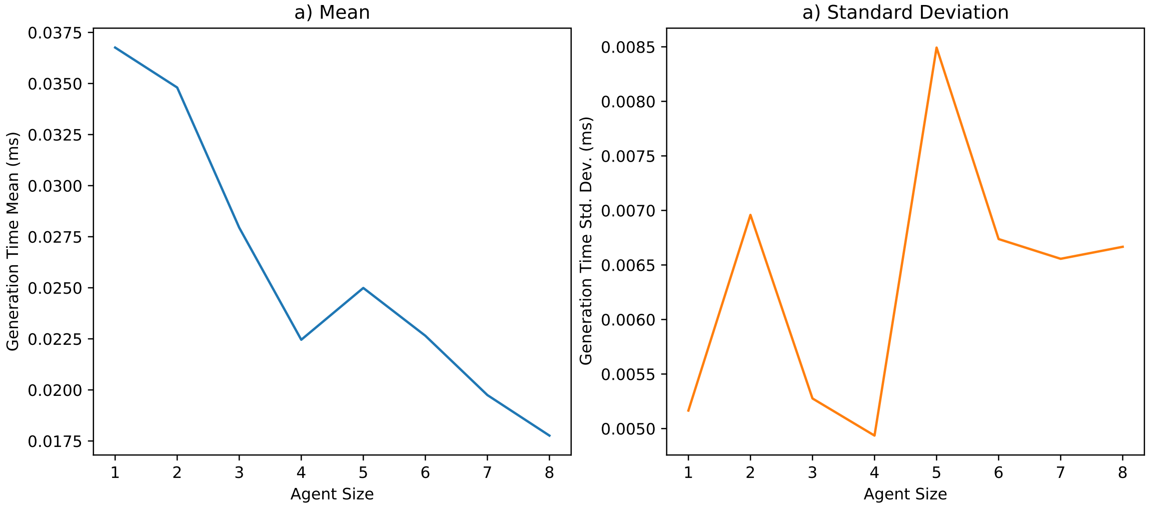


Figure 5.21: *Mean* and *Standard Deviation* for the *generation time* in milliseconds regarding a *second-order Markov Chains module* per agent size.

The system has 5 Markov Chain modules per agent, which means that these results needs to be multiplied by 5 and by the total number of agents working in the system to obtain the actual build or generation time.

Additionally, Figure 5.22 illustrates a segment of a user performance that indicates some generated material from the Markov Chain modules in a session of 8 agents. The blue crosses are generated *note on* messages and the red *Xs* represent every beat from the metronome. Note that the generated material try to be synchronized with the metronome in the same time point or in between, but still in some cases there is a small misplacement due to the way in which duration and silences are managed by the Markov Chains modules. Also note that, in this period of time, the user was changing the tempo since the beats are not equally spread.

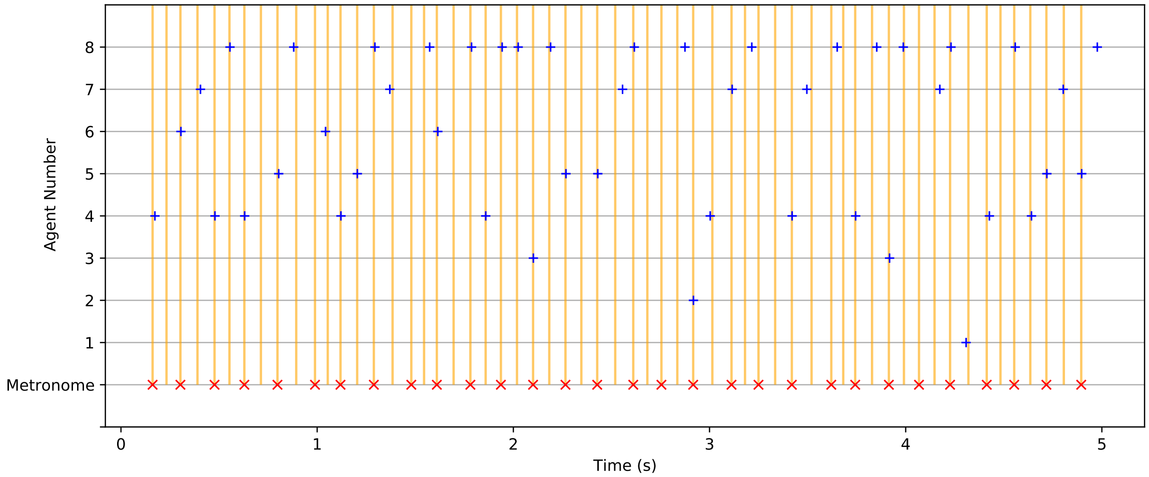


Figure 5.22: User session for 8 agents. The blue crosses are generated *note on* messages from the Markov Chains modules, and the red *Xs* represent every beat from the metronome.

Movement Algorithm

The frame time to process the autonomous movement is scheduled every 30 ms, nevertheless, the actual time can be affected by different aspects in the system. Table 5.13 shows the descriptive statistic for this parameter considering 2055 samples. Figure 5.23 depicts the corresponding boxplots in which a considerable amount of outliers is noticeable.

Agent Size	**Min.**	**Median**	**Mean**	**Max.**	**Std. Dev.**
1	20.662	29.954	29.979	42.115	2.274
2	21.627	30.034	29.982	39.250	2.427
3	18.504	29.909	29.986	40.709	2.560
4	18.247	29.994	29.988	42.981	2.843
5	10.761	30.014	29.999	69.607	3.804
6	9.380	30.064	29.987	48.712	4.607
7	16.534	29.944	29.989	43.687	3.383
8	12.218	29.797	29.989	51.929	4.902

Table 5.13: Descriptive statistics for the *frame time* in milliseconds regarding the movement algorithm per agent size.

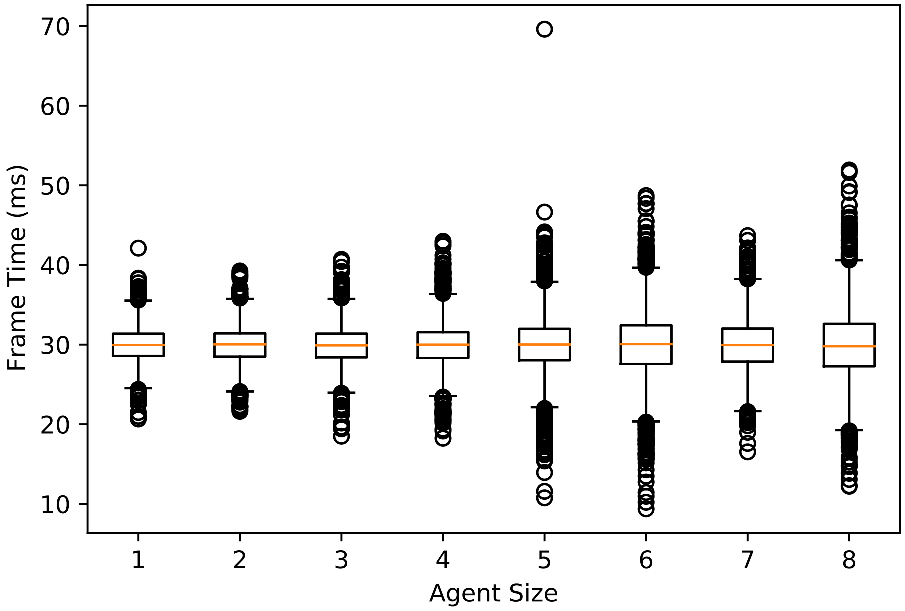


Figure 5.23: Boxplots for the *frame time* in milliseconds regarding the movement algorithm per agent size.

A Levene's test ($P < 0.05$) and a Kruskal-Wallis test ($p > 0.05$) were applied. It was found that there is a statistically significant difference in variances, but means are similar between groups. The found differences and the outliers respond primarily to the actual process that is happening in the scheduled task. That process is the calculation of the movement algorithm which is described in terms of the *processing time of the task* in Table 5.14 with boxplots shown in Figure 5.24 per agent size for

2041 samples.

Agent Size	**Min.**	**Median**	**Mean**	**Max.**	**Std. Dev.**
1	0.173	0.711	0.892	6.316	0.613
2	0.359	1.396	1.363	7.046	0.697
3	0.473	1.263	1.515	7.903	0.755
4	0.816	2.152	2.399	8.999	0.917
5	0.741	2.073	2.506	9.176	1.139
6	1.042	2.894	3.470	13.907	1.507
7	0.994	3.017	3.499	10.446	1.572
8	1.241	2.901	3.437	12.443	1.594

Table 5.14: Descriptive statistics for the *processing time of the task* in milliseconds per agent size.

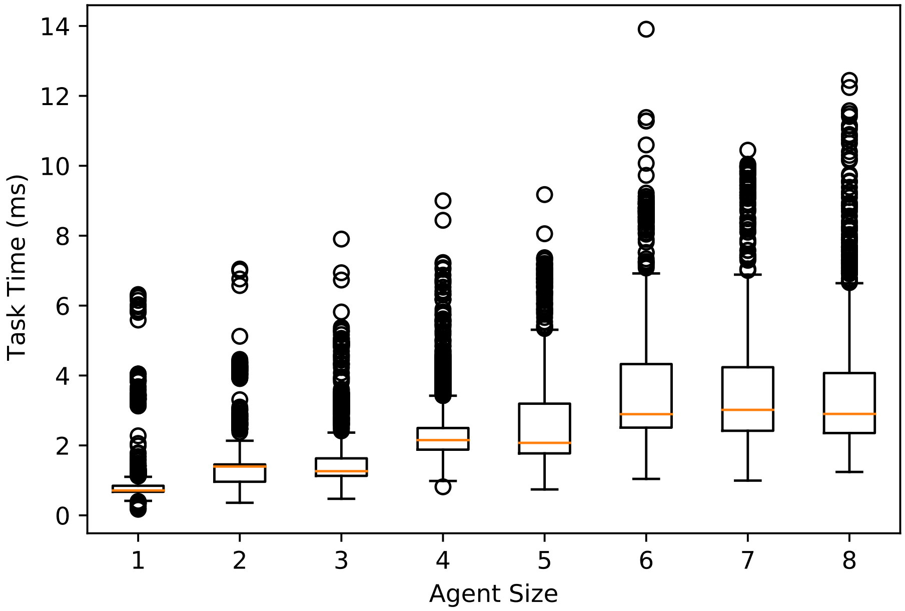


Figure 5.24: Boxplots for the *processing time of the task* in milliseconds per agent size.

It is evident that this parameter increases as agents are added, as well as the variance. Figure 5.25 illustrates this behaviour for both descriptors by showing how they are increasing.

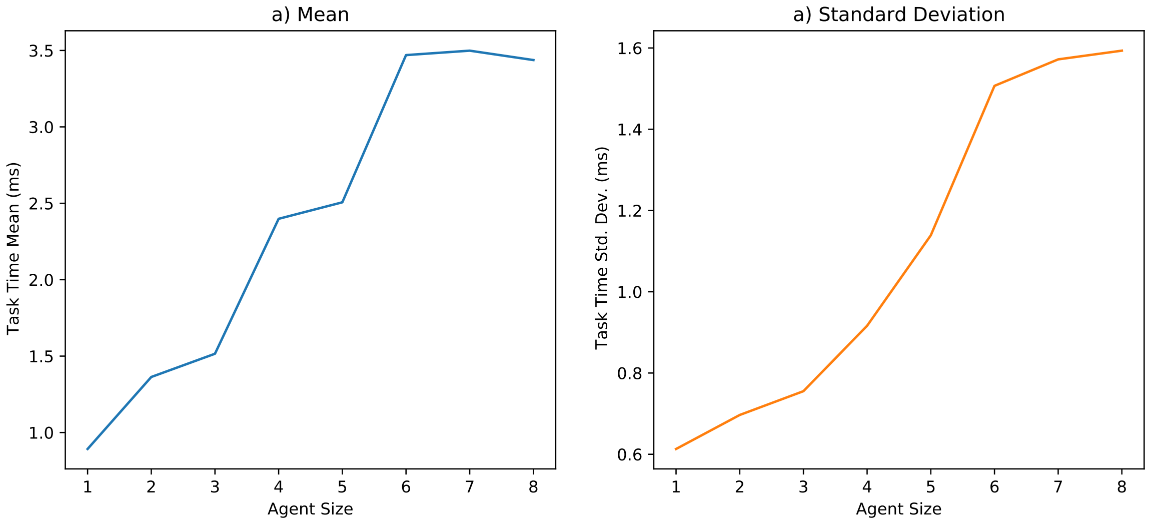


Figure 5.25: *Mean* and *Standard Deviation* for the *processing time of the task* in milliseconds per agent size.

5.3 User-Agents Interaction

For user evaluation, 7 participants took part of individual sessions to improvise a musical piece using the system and the infrastructure described in this thesis. The users are musicians with formal education that have had experience with looper devices, sound synthesis, and music improvisation, as well as common Extended Reality (XR) technologies (except the Microsoft HoloLens). Other details about their background is presented in the Appendix sections B.1 and B.2.

The first user is referred in this work as *User 0*, which was the pilot to improve the following evaluation sessions.

A session was organized in three parts:

1. Explanation about the system operation, HoloLens tutorial (included in the device), and HoloLens individual calibration.
2. Improvisation of a musical piece using the system.
3. A survey and reflections about the experience.

The sessions lasted more than one hour for each participant. The actual time spent in the music improvisation is shown in Figure 5.26 with a duration between 24 and 43 minutes.

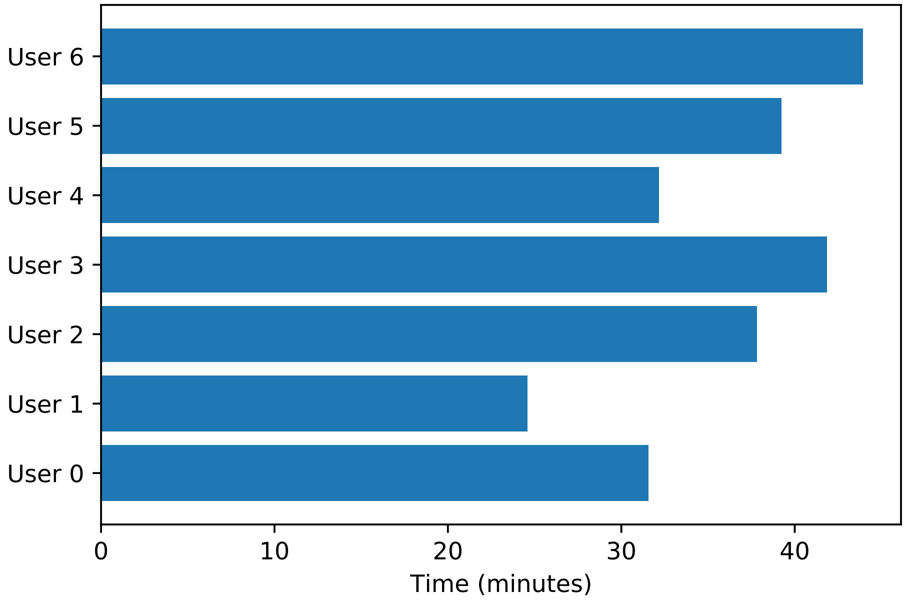


Figure 5.26: Session duration in minutes for each participant. The longest session lasted approximately 43 minutes, and the shortest one 24 minutes.

The captured data is totally anonymous since the recorded parameters, as described in section 3.5, do not allow any identification of the participants. The survey is anonymous as well, and the two data sources (*recorded file* and *survey*) are linked through a user ID given by a number (0-6).

Additionally, the audio output for every session was recorded from the 8 loudspeakers and then rendered in a binaural format that can be listened on the blog post[4] related to this thesis.

The results for every user regarding the captured data is illustrated in Appendix A. In this section, only one arbitrary user (**User 6** - last tested user) will be chosen to explain the results obtained so that it can be extrapolated to the other participants.

In the case of the survey, the results for all users are depicted in Appendix B. Some of these outcomes will be presented in this section and compared with the system measurements described previously.

5.3.1 Captured Data

The User and the Physical Space

The physical performance area where the user was capable of moving is a square of 6 by 6 meters approximately. The MIDI controller is placed in a table (height = 74 cm, width = 140 cm, depth = 70 cm) close to the center. With the position data from the HoloLens and the gaze direction, it is possible to know where the user was moving and to estimate where he or she was looking during the whole session.

Figure 5.27 shows how close or far was *User 6* from the center of the room. This user spent more time around 1.6 m away from the center and was sometimes 2.8 m away. This chart tells us that there were some points in which the user started to explore the behaviour of the system when moving around, but still near to the MIDI controller most of the time for the performance.

[4] https://mct-master.github.io/masters-thesis/2022/05/15/pedropl-human-machine-impro.html

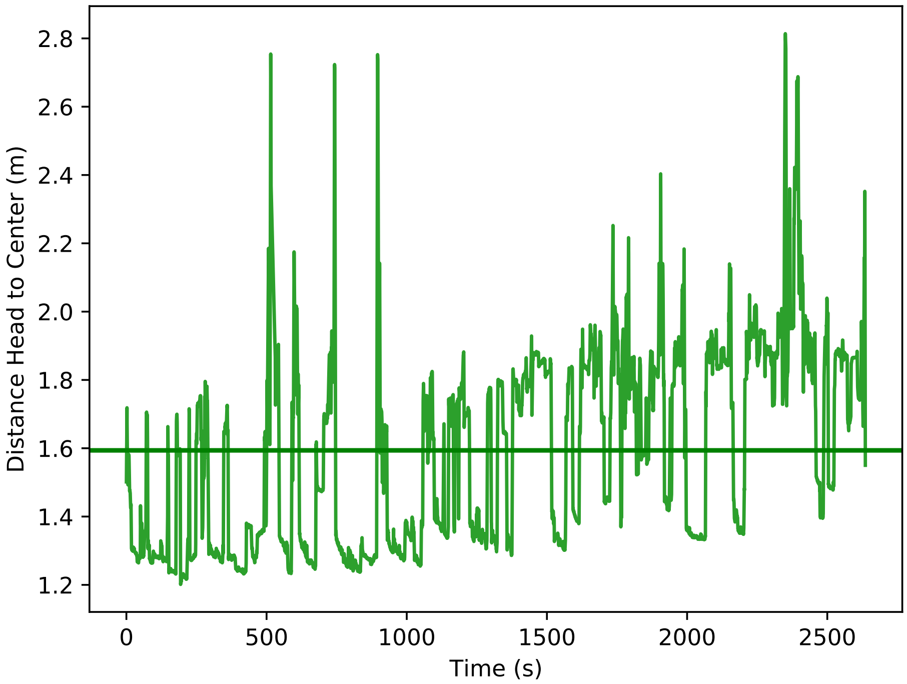


Figure 5.27: Distance (in meters) from the head (HoloLens) to the center of the room during the performance session for *User 6*

It is possible to check more clearly where the user spent most of his or her time by looking at the heatmap depicted in Figure 5.28. It confirms that the physical activity happens mostly around the center of the room. For *User 6*, if the hottest spot is not considered, we can additionally realize that he or she was a moving frequently around one meter in the back of this spot. For the back (x, z) view, the hottest spot can be interpreted as the point where the head was when the user was bent over, and the stain on top is when the user was standing straight, which confirms that most of the time he or she was focused on the MIDI controller.

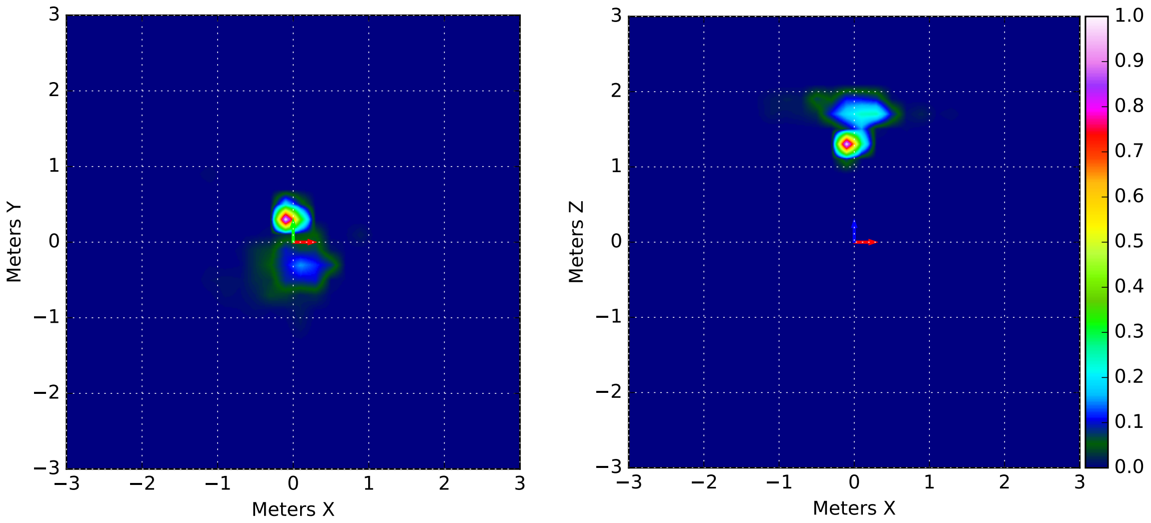


Figure 5.28: Locations where *User 6* spent most of his or her time during the performance session. This view is from top (x, y) and from back (x, z).

The gaze direction from the data determines the possible user attention during the session. It cannot be taken as a perfect indicator of the actual user gaze since it represents the forward vector of the HoloLens and it does not track the eyes movement. Figure 5.29 shows the azimuth and elevation angles for the gaze direction from *User 6*. In this case, this user is focused mostly in front and slightly below the XY plane (around -50°). The heatmap from Figure 5.30 confirms this behaviour. Moreover, there are points in time in which the user explores other angles, especially in the back and left side of the room as the graphs suggest. Furthermore, this results corroborate that the user was focused in the MIDI controller most of the time, since it is placed in front and below the performers' head.

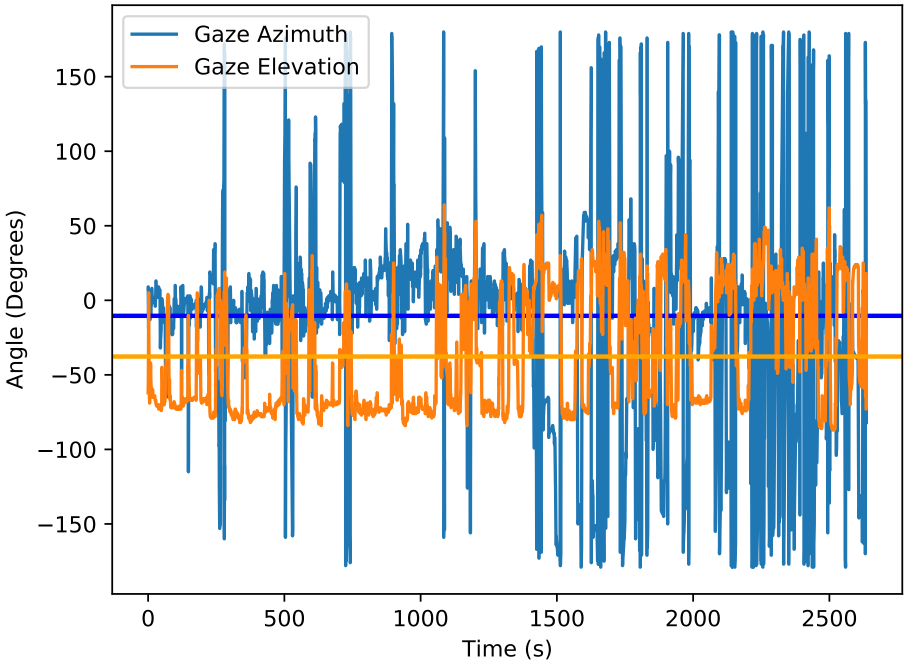


Figure 5.29: Azimuth and Elevation angles (in degrees) for gaze direction regarding *User 6* during the performance session.

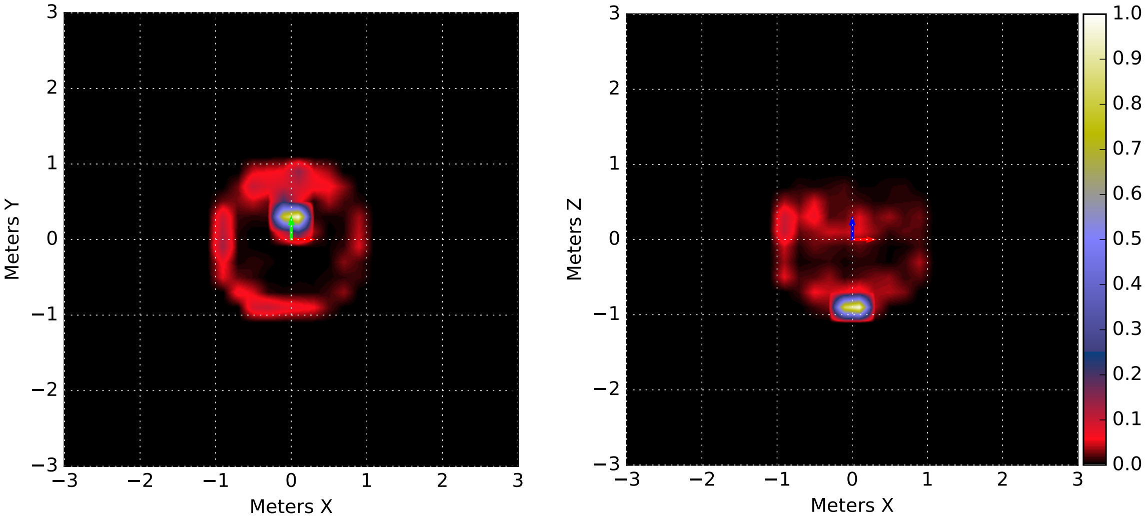


Figure 5.30: Directions where *User 6* spent most of his or her time during the performance session. This view is from top (x, y) and from back (x, z). As this is a unit vector, the planes are just a reference and does not represent the actual room and its center.

The User and the Rigid Body

During a session, the users had the freedom to place the rigid body (experiencing the 3D sound and visualizations) wherever they want around the room. They used mostly tables and small stools they found in the room. The tables were the same as the one used for the MIDI controller, and two of them were placed in the lateral sides. There were 5 stools of height 44 cm around the center, usually they were 2 meters far from the center over the floor. Some users put the rigid body on top of the tables, on the stools, or even moving the stools over the tables and placing the rigid body in a higher position.

These placement practices are reflected in Figures 5.31 and 5.32 for *User 6*. The first plot describes the distance between the performer's head and the rigid body along time, with periods in which the distance tends to be constant (when the rigid body is static in one only place). The other image is a heatmap that clearly denotes the positions where the user left the rigid body during the session. Note that in the back (x, z) view, there are different Z values since this user tried to experiment with several heights.

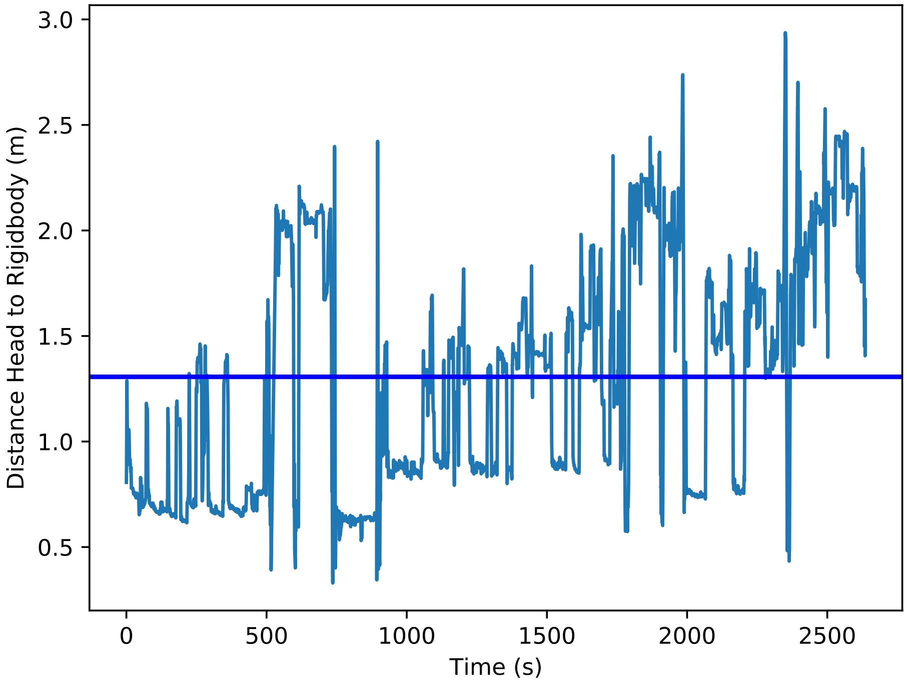


Figure 5.31: Distance (in meters) from the head (HoloLens) to the rigid body during the performance session for *User 6*.

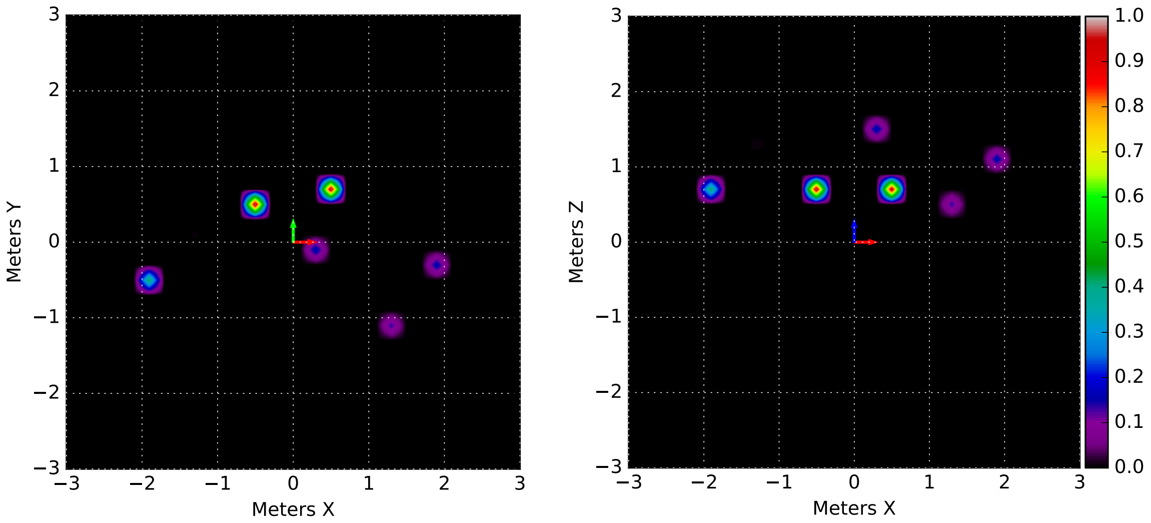


Figure 5.32: Frequent locations where *User 6* placed the rigid body during the performance session. This view is from top (x, y) and from back (x, z).

The User and the Agents

The distance between the performer and each agent during a session is shown in Figure 5.33 for *User 6*. Note that, as the time advances, the agents increase their distance at some points. Besides, sometimes they crossed the zero distance since they can pass from one point to another trough the user and no just moving around. The *users' movement*, *gaze*, and the *awareness of the agents with each other* contribute for this behaviour.

Figure 5.34 illustrates how the distance from the *user* to the *center of the agents' swarm* evolves along the session. For *User 6*, the are usually one meter far from the head as a swarm, although sometimes they went far and came back, specially at the end since the user tends to explore the environment when all agents are moving autonomously.

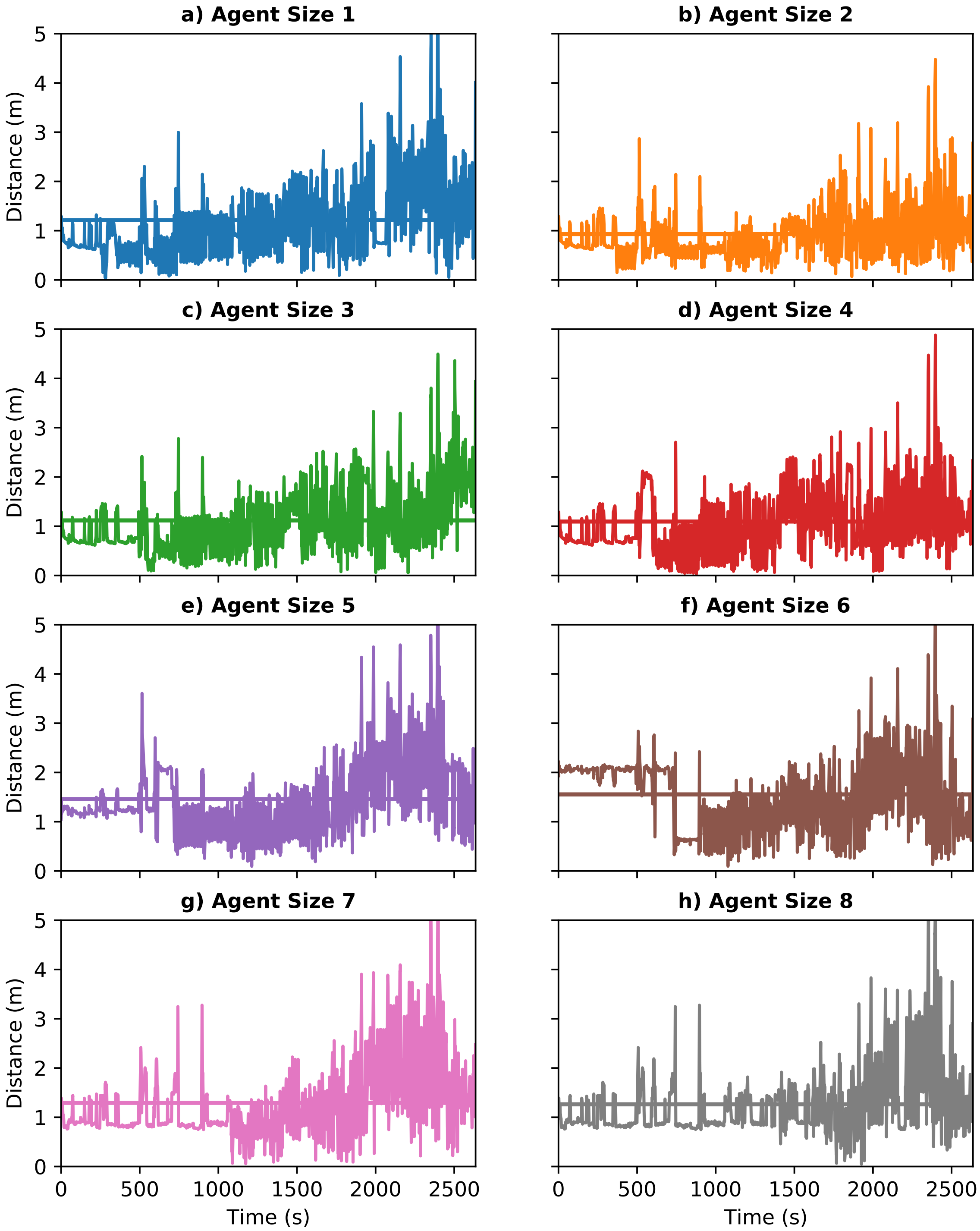


Figure 5.33: Distance (in meters) from the head (HoloLens) to every agent during the performance session for *User 6*.

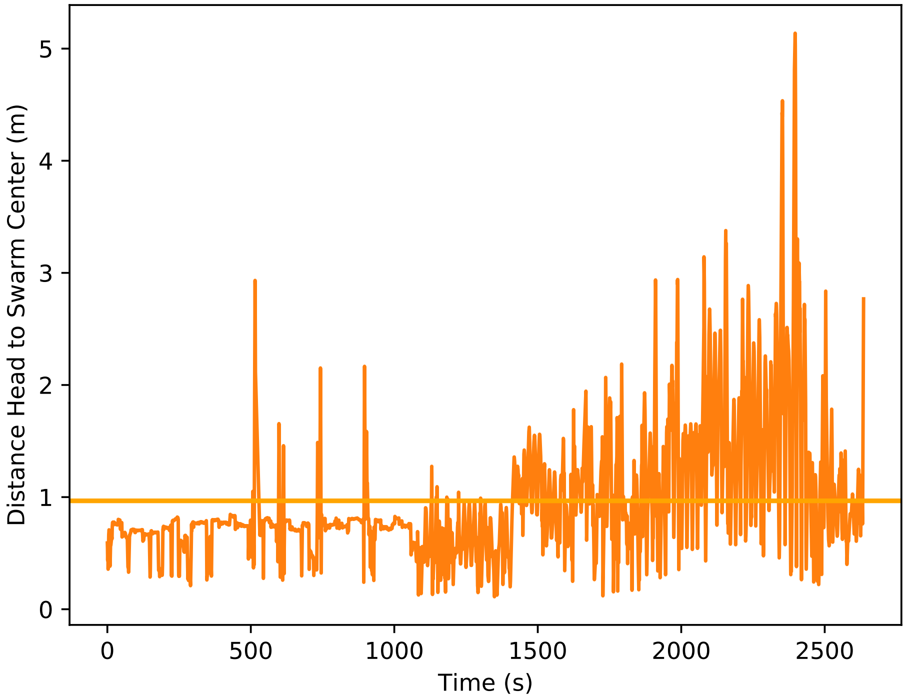


Figure 5.34: Distance (in meters) from the head (HoloLens) to the agent's swarm center during the performance session for *User 6*.

The attention to the agents can be provided by checking if they are observed by the user. The closest estimation for this parameter can be obtained when the agents are in the *field of view* (*fov*) of the HoloLens. Figure 5.35 depicts how the agents appear in the field of view for *User 6* compared against the user activity in the MIDI controller. This plot indicates when the agents starts to appear for the first time with corresponding arrows per each row. Note that most of the *fov* activity happens almost in the end of the performance, coincidentally in the period when the user is not active in the MIDI controller, it means that he or she is exploring the environment and inspecting what the agents are doing in terms of music material and movement.

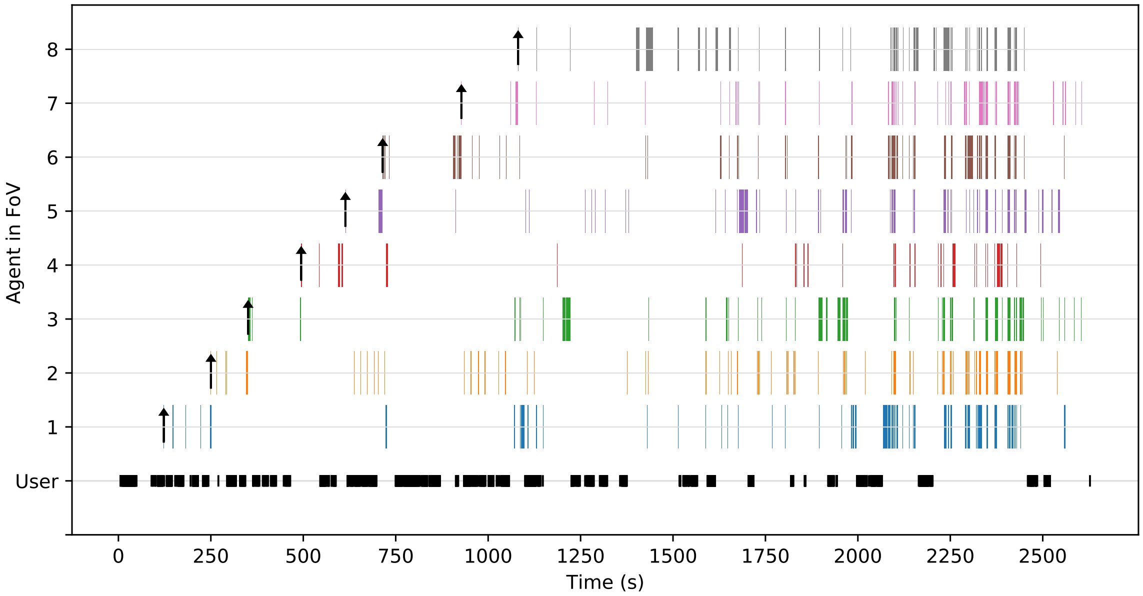


Figure 5.35: Periods when agents were in the HoloLen's *field of view* (fov) during the performance session for *User 6*. The first row 'User' represents the user activity in the MIDI controller along the session. The arrows represent the moment of the first appearance of an agent.

The periods in which a user actually has control of an agent is when he or she *locks* one of them. These periods can be appreciated in Figure 5.36 for *User 6*. At the beginning, the user built every track up to the maximum allowed (which is eight), then there are some periods when he or she *locked* or *released* the agents. Note that every agent spent most of the time in 'RELEASED' mode since the user can control only one at a time. There are time frames in which there is no activity in the MIDI controller and no agent is locked, thus the user was observing and possibly moving around the room to look at the environment.

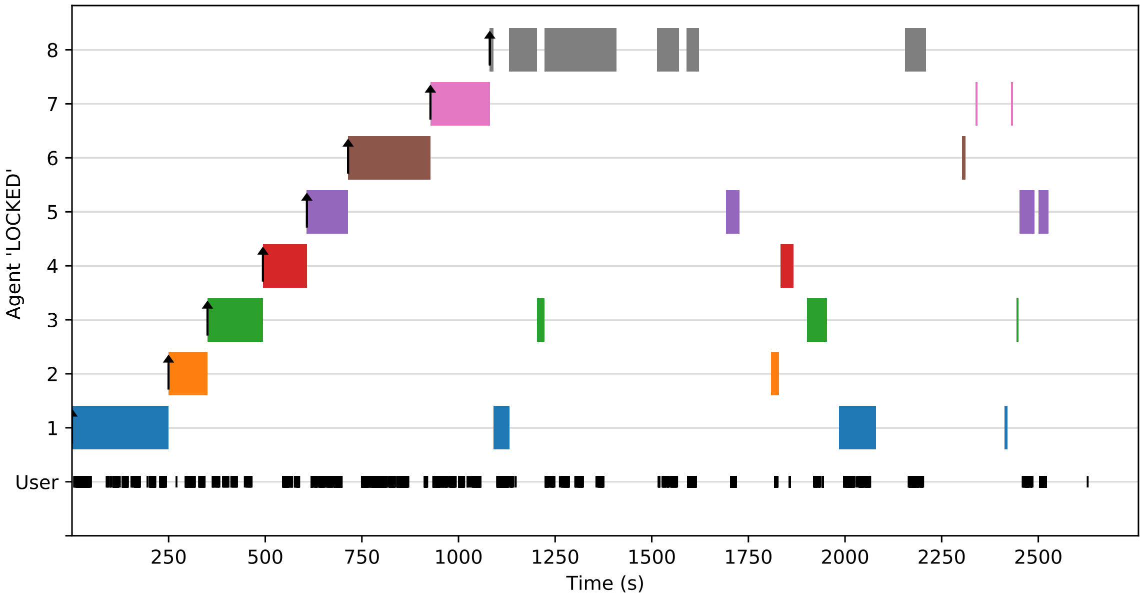
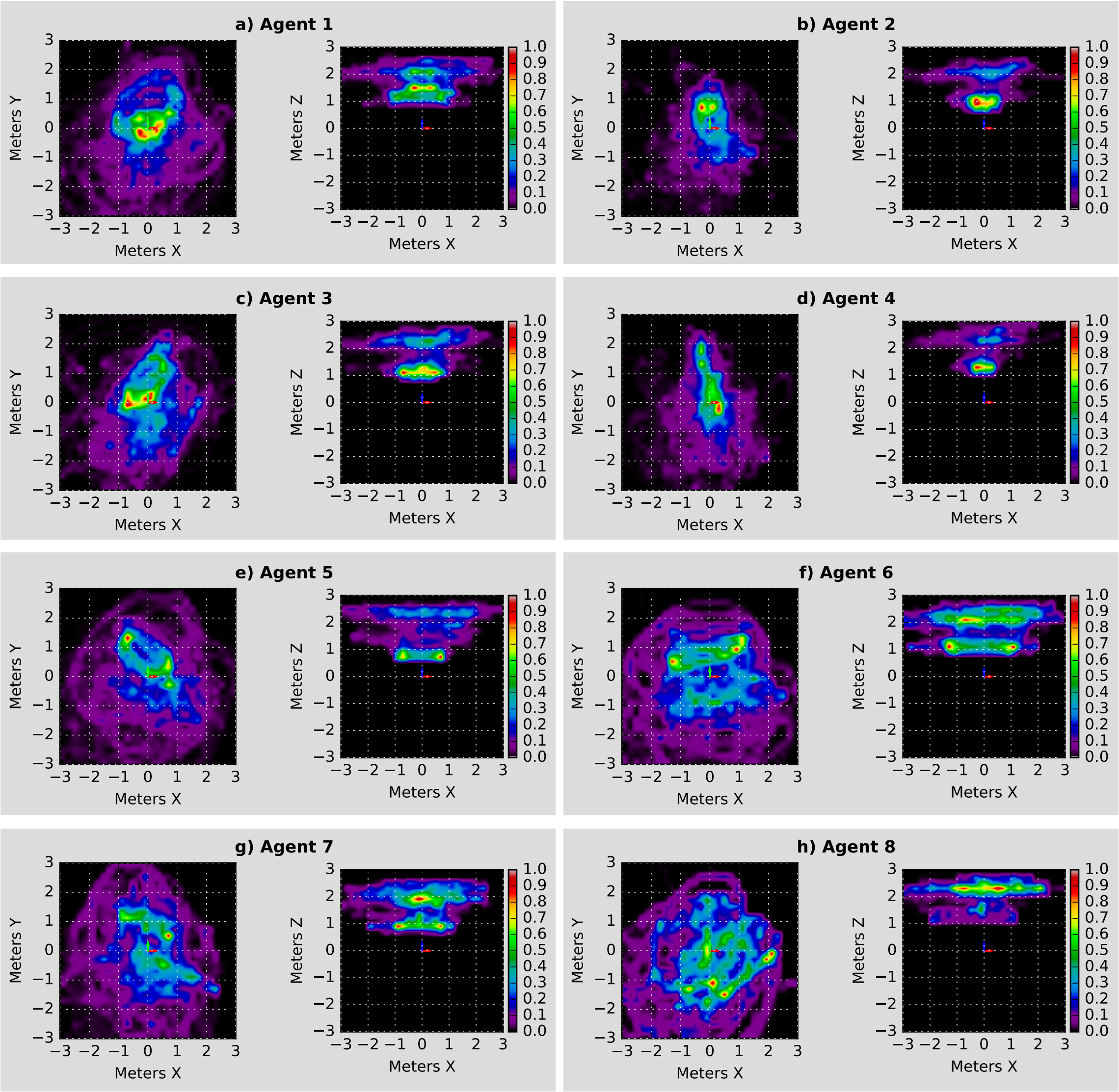


Figure 5.36: Periods when agents were 'LOCKED' during the performance session for *User 6*. The first row 'User' represents the user activity in the MIDI controller along the session. The arrows represent the moment of the first appearance of an agent.

5.3.2 Agents Movement in a Session

Part of the autonomous movement behaviour executed by the agents was shown previously in Figure 5.33. It takes into account the distance between the HoloLens and every agent for *User 6*. Additionally, Figure 5.37 illustrates a set of heatmaps per agent for the same session performed by *User 6*. Each graph denotes the time spent by the agents in several locations through the room from top (x, y) and back (x, z). Note that some trajectories can be identified and different movement patterns are found. The fact that these trajectories are rendered in a heatmap demonstrates that an agent travels the same path (at least in the same plane) for a considerable amount of time (or in different intervals during the session) before changing to a new route, which may influence the movement predictability from the user.


a) Agent 1

b) Agent 2

c) Agent 3

d) Agent 4

e) Agent 5

f) Agent 6

g) Agent 7

h) Agent 8

Figure 5.37: Heatmaps for the movement of every agent during the performance session for *User 6*. It shows the locations where they spent most of their time, as well as trajectories. The view is from top (x, y) and from back (x, z) per each agent.

5.3.3 Manual and Autonomous Music Generation

In the previous section, the user's activity in the MIDI controller was compared against the agents' *fov* and *'LOCKED'* periods during the performance for *User 6* in Figures 5.35 and 5.36 respectively. This activity includes any MIDI message fed by the user such as *note-on*, *note-off*, and *controls* for the looper and sound synthesis parameters. It can be considered as the user participation for the overall music improvisation.

From these inputs, there is a special emphasis in the note-on and note-off messages, since they are the training material for the *Markov Chains* modules that integrates the autonomous music generation. The maximum amount of these messages when a user records a loop line is displayed in Figure 5.38.

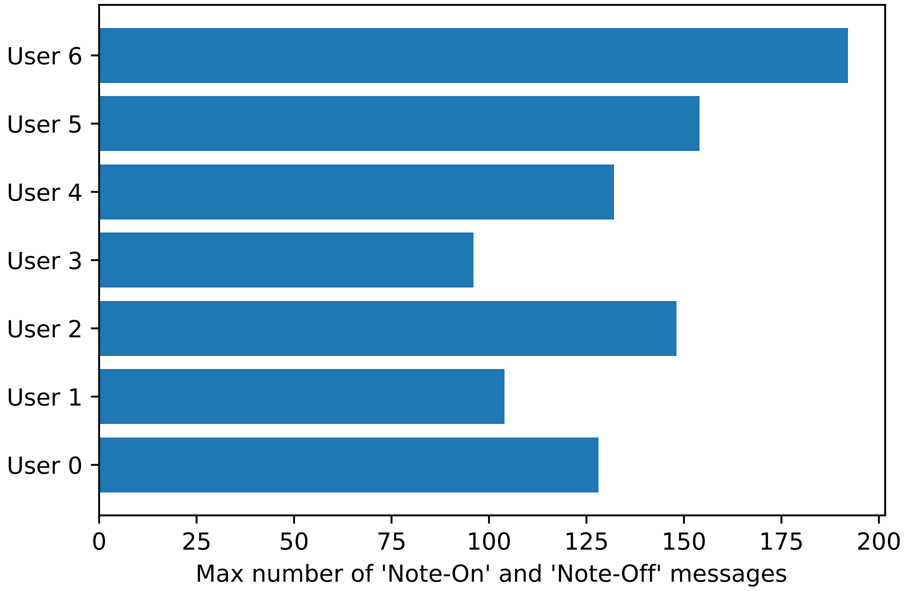


Figure 5.38: Maximum number of MIDI messages (note-on, note-on) that were entered by each user during their performance sessions. These values are calculated considering every recording period from the looper.

If we take the maximum value from the previous chart, which is 192 messages per recording line, and multiply it by 5 (number of Markov Chains modules in the system), we obtain 960 samples to train the modules. We can calculate an estimated build time by using the linear model found in section 5.2.4, the result is *2.24 ms*. This is not a high precise estimation since the modules contributes with different parameters, but at least it gives us the idea that the build time is not a considerable issue when it comes to a real-time setting developed under the design of this system.

Tempo Changes

The system allows the user to change the musical tempo. This changes influence in the speed of the generated music material since it is synchronized with the metronome. Figure 5.39 shows the change in the beat duration of the metronome along the performance time for *User 6*. This chart is limited to periods of 5000 ms. The high impulses correspond to periods where the user changed the tempo or just stopped the metronome, wait for a moment, and played it again. We notice that this user changed the tempo 4 times during the performance and the second to last change has the faster metronome since it portrays the shortest beat duration.

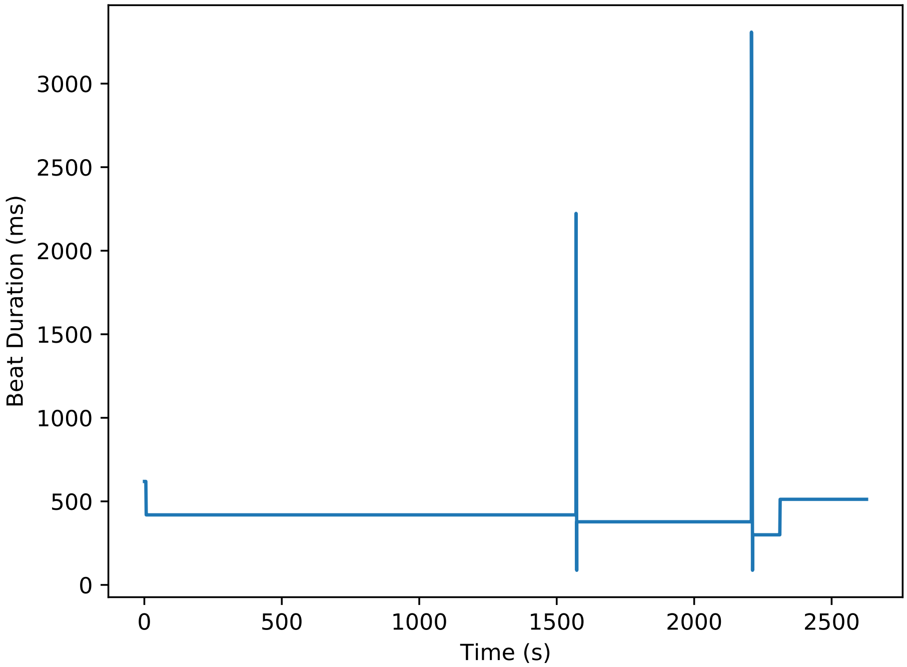


Figure 5.39: Beat duration period for the metronome during the performance session for *User6*. This is limited to periods of 5000 ms to detect the actual BPM configuration given by the user at any point in time. The high impulses correspond to periods where the user changed the tempo or just stopped the metronome, wait for a moment, and played it again.

5.3.4 Survey

The survey results are shown in Appendix B. It was given immediately to the participants after their music session, except for *User 0*, who received the survey one day after. It is organized in three parts: *Musical background*, *Extended Reality (XR) technology experience*, and *music session experience*. The first two parts are described at the beginning of this section 5.3, and some questions of interest from the last part are analyzed below having as reference a system working with 8 agents.

Latency and Jitter

The system with 8 agents has a *MIDI Controller to Sound Output* latency of **44.46 ms** and a jitter of **6.96 ms** as shown in section 5.2.1. The latency perception from users was asked in the survey giving the results illustrated in Figure 5.40.

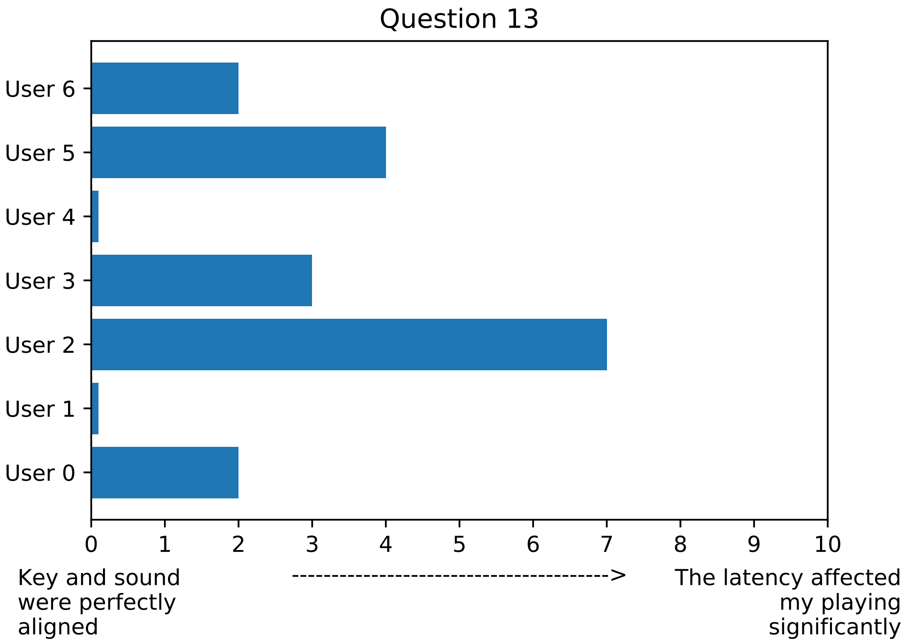


Figure 5.40: Question 13: Fluency for playing: Did you feel a significant amount of latency (a delay time between the moment you play a key and when you hear the generated sound) when performing?

We can notice that users actually felt certain amount of latency, but it was not a significant issue when playing, except for *User 2*. We can speculate that it happened because he or she increased the tempo considerably at some parts of the performance as illustrated in Figure 5.41.

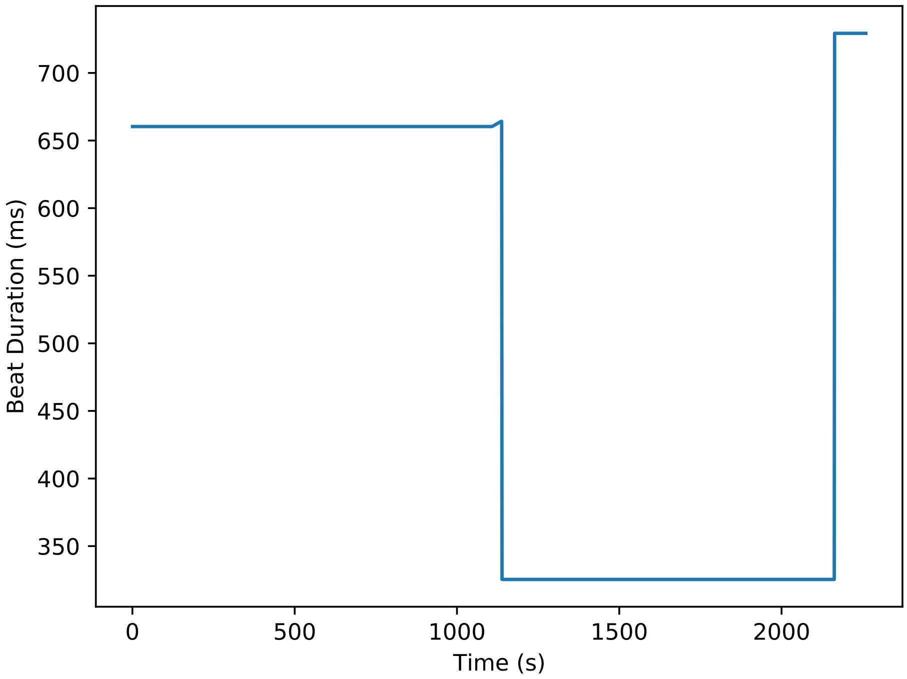


Figure 5.41: Beat duration period for the metronome during the performance session for *User 2*. This user changed the tempo three times in which the second change has a fast tempo of 184 BPM (beat duration = 325 ms) approximately.

The *Spatial Audio Placement latency* for 8 agents is around **134 ms** and the *Sound to Visualization latency* is **6.747 ms** with a jitter of **13.44 ms**, as well as a *packet loss* of **23.89%** for the HoloLens communication. Despite of these values, the results for questions presented in the Appendix sections B.3.3, B.3.4, and B.3.6 show scores close to 8 out of 10 in terms of perceiving an alignment between rigid body, sound, and visuals. It can be explained due to the human limitation to identify directional sounds as stated by Mills (1958).

In contrast with the results mentioned above, the human limitation could also affect the capacity to identify the location of sounds since the question related with this aspect in B.3.5 was given 5 out of 10 in average, especially when there are many of them moving around a user. However, the visual confirmation from the HoloLens helped to localize the agents as a multimodal integration for the interaction. Question B.3.7 illustrates that agents are not that difficult to identify visually (7 out of 10).

Music Improvisation

Regarding the use of the looper through the MIDI controller, users reported a score close to 8 out of 10 in B.3.14. From the performances' observations, they did not struggle much with the controller since it is a familiar device according to their musical background. Additionally, they manipulated the sound synthesis parameters considerably, since in B.3.19 they gave 9 out of 10 about how much they changed the sound properties per track.

Questions from B.3.15 to B.3.18 describe the human-machine music interaction from the users point of view. From these questions and their reflections, they felt that they were not in total control of the improvisation, some of them requested a mechanism to change between the autonomous and the manual composition, other decided to adjust to the machine intention. Moreover, they partially agreed with the material generated by the agents (5 out of 10) and believed, to a certain extent, that the agent's lines were close to their musical vision, although they felt that the generated material was a bit out-of-sync regarding the metronome (5 out of 10).

Autonomous Movement

Questions from B.3.10 to B.3.13 show the results regarding the movement behaviour for the agents. Users perceived the agents' speed a bit fast in some occasions and noticed that they were following them. Furthermore, they could predict in some sense the movement patterns, but not at all (6 out of 10), and instead of being equally spread around them, they reported that this entities looked like a school of fish that tended to go to certain positions as a swarm. Some users thought that they were dancing and were coordinated between them.

This flocking behaviour can be described in terms of the *distance to the center of the swarm* as in Figure 5.34 from the previous section. Likewise, the heatmaps presented before in Figure 5.37 show that it is possible to guess some of the agent's path if they are observed thoroughly, although they also vary their movement during a session.

Mixed Reality Experience

During a session, some users experienced issues with the HoloLens regarding the *selection gesture*. Sometimes they tried to select a sphere but it did not respond on the first try. That is why they gave around 5 out of 10 to this interaction aspect and referred this problem in their comments. In addition, some of them found the device uncomfortable after some time, but others forgot this issue since they were immersed in the experience (in average, they rated 6 out of 10 for comfortability).

Another limitation from the HoloLens is the *field of view* (*fov*). The user is constrained to a little screen through the glasses and it is not possible to observe the augmentations around the device's periphery. However, the users did not feel significantly restricted since this issue was rated as 4 out 10 in B.3.21 about feeling limited by the HoloLens *fov*.

5.3.5 Overall User Experience

Regarding the musical piece produced by the improvisation session, users rated the overall sound mixing as 7 out of 10 in B.3.24, and the aesthetics as 6 out of 10 in B.3.24, both are average values. Besides the number of agents, users commented that 8 is a good number for some of them, others started to feel overwhelmed with that number, and one of them suggested to increase the number to 15.

Two important general and subjective questions were asked regarding the easy-of-use and the enjoyment. These results are depicted in Figures 5.42 and 5.43 respectively.

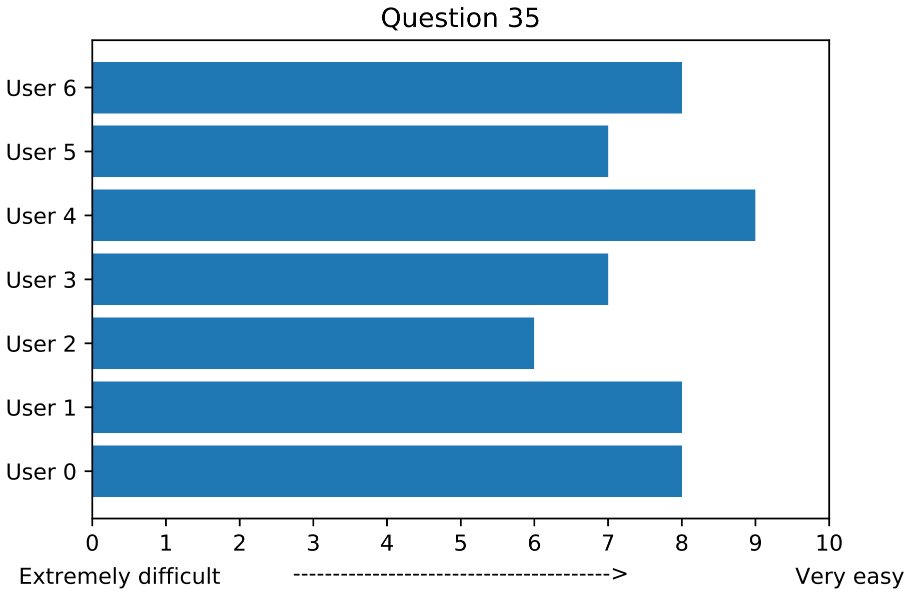


Figure 5.42: Question 35: Overall experience: In general, how easy was to use the whole system?

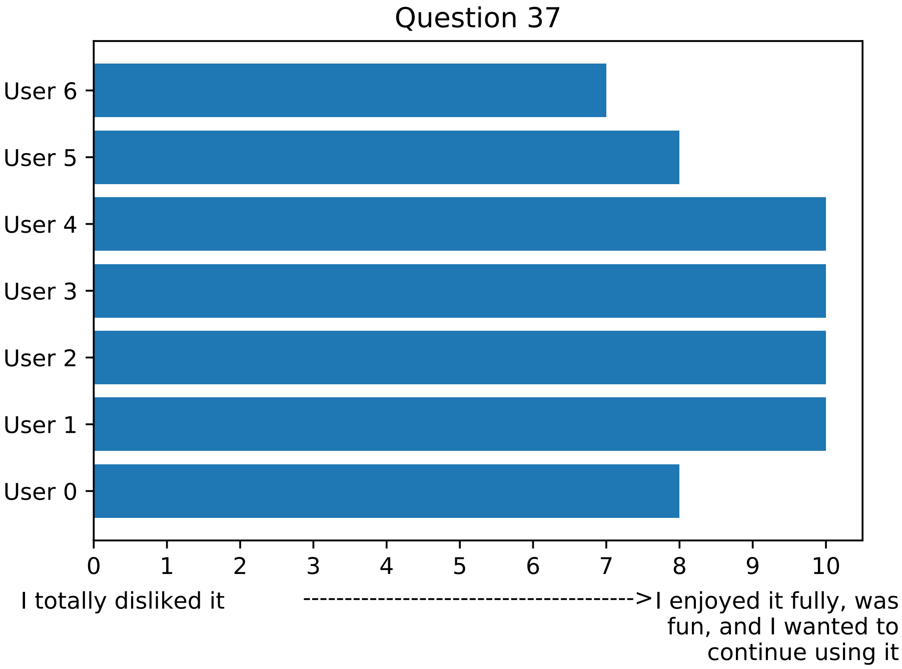


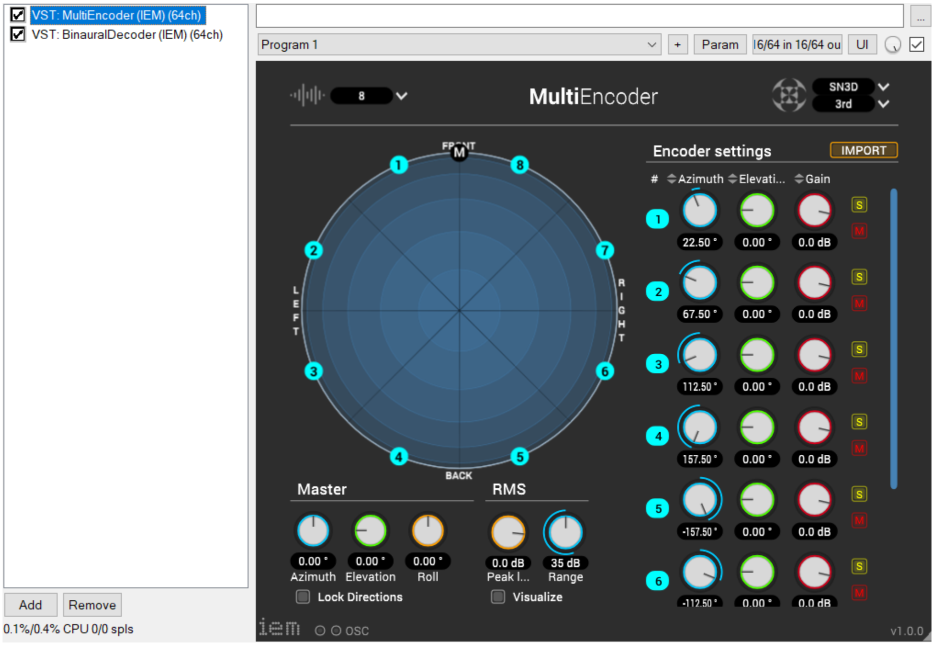
Figure 5.43: Question 37: Overall experience: How much did you enjoy the performance?

As a final reflection, they pointed out that it was a fun experience for a non-conventional music composition. Additionally, they provided the following suggestions to improve or change the experience:

- Explore different controllers to replace the traditional MIDI keyboard.
- Provide an integration with the hardware/software tools that a user is accustomed to using.
- Provide more freedom so that the session is not dependant of the metronome.
- Allow the user to keep or discard the music material from agents if needed.
- Improve the precision for the interaction with agents through the HoloLens.
- Find a way to link visually the spheres with the sound synthesis identity. When there are many agents it is difficult to remember the corresponding number to the sound.
- Provide a way to have a view of all the agents in a mixer to balance the volume.

5.3.6 Sound Recordings

The audio output from the music sessions were recorded directly from the 8 signals received by the loudspeakers array. The software *Reaper*[5] and the ambisonic plug-ins from *IEM*[6] were used to render a binaural version (2-channel) per each performance. Figure 5.44 shows the configuration for the transformation from 8-channels to 2-channels in the mentioned software.


☑ VST: MultiEncoder (IEM) (64ch)
☑ VST: BinauralDecoder (IEM) (64ch)

Program 1 | + | Param | I6/64 in 16/64 ou | UI

MultiEncoder — 8 — SN3D — 3rd

Encoder settings

#	Azimuth	Elevati...	Gain
1	22.50 °	0.00 °	0.0 dB
2	67.50 °	0.00 °	0.0 dB
3	112.50 °	0.00 °	0.0 dB
4	157.50 °	0.00 °	0.0 dB
5	-157.50 °	0.00 °	0.0 dB
6	-112.50 °	0.00 °	0.0 dB

Master: Azimuth 0.00 °, Elevation 0.00 °, Roll 0.00 °
□ Lock Directions

RMS: Peak I... 0.0 dB, Range 35 dB
□ Visualize

0.1%/0.4% CPU 0/0 spls

Figure 5.44: IEM Multi-Encoder for translating the 8-channel signal from a music performance to a 2-channel binaural format. The sound sources are configured considering the loudspeakers positions in the room.

The sound recordings are included in the blog post[7] associated to this thesis.

[5] https://www.reaper.fm/

[6] https://plugins.iem.at/

[7] https://mct-master.github.io/masters-thesis/2022/05/15/pedropl-human-machine-impro.html

5.4 Summary

This chapter detailed results and descriptions on measurements and user evaluation to validate the system proposed in Chapter 4. The implementation was tested for several number of agents regarding latency and other real-time parameters.

The system is designed to be *scalable* to any number of agents, nevertheless, limitations in computational power in the implementation led to choose a number of 8 agents for all the testing cases applied in this work. This *scalability* property allows to increase or decrease this number according to the available resources.

Table 5.15 shows the measurement values for a system working with 8 agents. These are the parameters that users had to deal with in their performances.

Parameter	**Value**
MIDI Controller to Sound Output Latency	44.46 ms
MIDI Controller to Sound Output Jitter	6.96 ms
Spatial Audio Placement Latency	134.892 ms
Sound to Visualization Latency	6.749 ms
Sound to Visualization Jitter	13.44 ms
Packet Loss System-HoloLens	23.89%

Table 5.15: Summary of system measurements for an agent size of 8.

The user testing sessions demonstrated that the previous values do not produce significant issues, except for the *MIDI controller to sound output latency* when a high tempo (BPM) is set. Moreover, the system was *stable* on every performance and tolerated a long session of 43 minutes.

Comparisons were established between the system measurements, the captured user data, and a survey applied to the participants. In general, the system provided to the users a positive unconventional experience that was considered enjoyable.

For a view from different perspectives regarding a human-machine performance using the system, the author recorded a video of himself using the system. This video can be found in the complementary material from the blog post[8] on the MCT website.

[8] https://mct-master.github.io/masters-thesis/2022/05/15/pedropl-human-machine-impro.html

Chapter 6

Discussion

This thesis explored ***how*** a *system for human-machine live music performances* can be *designed and implemented* in a *multimodal environment*.

As such exploration has a high number of potential solutions, a systematic literature review presented in Chapter 2, as well as criteria from the author's expertise, were employed for stating the system described in Chapter 4.

The objectives for this research work were met through the methodology established in Chapter 3. That is, a multimodal system for human-machine music improvisation was developed and evaluated under the concepts and paradigms related to *Musical Agents*, *Live Algorithms*, *Swarm Intelligence*, and *Musical XR* by including *motion capture*, *spatial audio*, and *mixed reality* technologies under the scope defined in the requirements presented in Chapter 4.

The evaluation that was carried out examined the *effectivenes* and *efficiency* of the system implementation from objective measurements, which revealed advantages and limitations for specific hardware and software frameworks.

Additionally, the system was tested by several users whose experiential data was collected in two ways: A recording of system events during a whole improvisation session, and the application of a survey after the end of the session.

The results obtained from these evaluations are presented and described in the previous chapter and will be interpreted and explained here, as well as the significance and implications of these findings.

6.1 Key Findings

Under the equipment and tools used for the implementation, the objective data from the system measurements suggests that the *number of agents* influences in the amount of *computational resources* required and the *quality of the sound output.* Moreover, the analysis identifies that the sound *latency and jitter* is relatively high and tends to be constant when a musical note is played regardless the number of agents, but other latency categories are prompt to increase as agents are added as well as the *packet lost* for transmitting data to the mixed reality headset. This implementation tolerates 8 agents before a significant degradation of the sound quality. Thus it was the limit for conducting user performance sessions.

The user sessions demonstrated the system *stability* at least up to 43 minutes, which also proved its *effectiveness* since it worked as expected. In this sessions the users explored the system capabilities as much as they could for a long period and *enjoyed* the session in general. There were interaction problems regarding the execution of the "air tapping" gesture for some users, but they managed to get used to them after a while. The latency was not a significant problem unless the musical tempo was increased significantly, and other latency parameters were not disruptive to the experience despite of being theoretically high.

These findings support the way-of-making for the system and confirm, based on this constraints, the feasibility of the solution under the proposed design.

6.2 Results Interpretation

6.2.1 System Measurements

Several types of latency values were measured in regards to the number of agents. The first was the delay between pressing a key in the MIDI controller to the sound output, which is a usual metric for an interactive music system (IMS). For this implementation, it was found that this latency and its jitter tend to be constant for any agent size. The values for this parameters, 44.46 ms and 6.96 correspondingly, are considered high, since a usual target for digital instruments is 10 ms with 1 ms of jitter according to Wessel & Wright (2002). However, other studies suggests that this value can be higher without affecting the experience significantly.

Bartlette et al. (2006) set a threshold of 86 ms for network performances in collaborative settings, and Schuett (2002) sets different values according to the tempo, in which one of then is 50-70 ms when performers try to use a copying strategy in a performance. This can be contrasted with the user sessions where they don't report major problems for this latency parameter, except for one participant. It was found in the recordings that this particular participant increased the musical tempo to 184 BPM at some point, which might have affected the latency perception.

In the case of the *spatial audio placement latency*, we have a high value of 134.892 ms when 8 agents are instantiated. It indicates that, when the *spatial positioner* is moved to one direction, we hear the sound after this time. The users did not report a significant misplacement of a sound in terms of spatialization. According to Mills (1958), we are limited to perceive sound direction in certain angle ranges depending on the frequency, which degrade the human precision and allow to think that the sound is placed correctly in the intended position. Furthermore, the measurement technique for this case did not have a high resolution, thus it is possible that this latency value is +/- 16.67 ms (60 fps from the camera that measured).

For the visualization latency regarding sound for the same 8 agents, we have a time of 6.96 ms, which is a good value and unexpectedly low. It is low because of the audio latency for spatialization. Besides, the users responded positively to the match between the spheres and the sound. However, we have a high jitter value of 13.44, which can be produced by the packet loss of 23.89% between the system and the HoloLens. Possible explanations for this value is that the connection is over WIFI, which increases the instability when there are more signals around and the configuration of the room influences in the reflection of the wireless transmission.

Other values from the results has a minimal or negligible influence in the performance of the system.

6.2.2 User-Agents Interaction

The user performances were longer than expected (15 min was planned). The participants used the system between 24 and 43 minutes. In that time, the system was stable without serious interruptions or crashes. One reason for these times could be the *novelty* of the system and their music backgrounds, since most of them has had experiences with loopers and sound synthesis.

In the results there are several charts regarding user and agent movements. For the case of users, the analysis identifies how they were interacting with agents and the physical space. All of them spent most of its time close to the MIDI controller and usually they started to explore the environment after using the maximum of 8 agents. It suggests that the users wanted to be constantly active in the performance and no just looking around for long periods.

Despite of this user behaviour, charts related with distances between the user and other objects reveal that there were moments in which users moved away from the center to explore the space, maybe *caught* an agent, and came back to the MIDI controller for more direct music-making. They were driven by curiosity for trying all the possibilities that they could, according to comments after the performance.

One interesting aspect is that they used the environment as part of the system, for instance, they moved tables and stools that were found in the room to place the *spatial positioner*. This act is reflected in the heatmaps that correspond to the *rigid body* placement.

Regarding the agents in a performance session. It was reported that they were moving like a school of fish sometimes, and tend to be in one side of the room depending of user position. Moreover, some users claim that they could predict the movement to some extent, others that it was not possible. This is due to the amount of attention to the agents. The resulting behaviour actually allows that an agent travels the same path for a period until some parameters change. it is confirmed in the heatmaps per agent that were presented in the results.

The other part of the human-machine interaction is the *music generation*. Most of the users reported that they did not agree in losing totally the vision of the performance when an agent changed the musical material, but still they thought that the machine followed their style at some degree. One of the users explained that he tried to follow the machine instead of forcing the machine to follow him or her. This behaviour brings to a debate regarding the level of autonomy in a human-machine music performance.

Regarding the mixed reality interaction. Some users struggled with the "air tapping" gesture. It could have be caused by a bad calibration, errors in the MR application, or lack of practice in such environments even in the case of a relatively simple gesture. Moreover, some users reported uncomfortability after a while and some minor limitations for visualizing the agents because of the HoloLens field of view. These are common issues that are found in works related with mixed reality that used the HoloLens, as mentioned in Chapter 2.

In general, the users rated the system as an easy-to-use music platform and express that they enjoy the session in a high degree, which explains the long times they spent in the performances.

Regarding the audio recordings, users were asked to rate the aesthetics of their work. The lower score was 4 and the highest 8 out of 10. It is noticeable that they are quite different pieces of music, even considering that there were predefined sounds and presets. It confirms, together with the data recorded, that users made the effort to change synthesis parameters during a session, which could be seen as an effort to give an identity to the composition.

6.3 Limitations of the Study

The first limitations to take into account are the availability of resources and technological constraints. For instance, the spatial audio does not provided an actual 3D soundscape because more loudspeakers are needed in different directions; the HoloLens has a limited field of view, poor ergonomics, and needs to be charged eventually; the motion capture has limited detection area, and the computer for the system was not powerful enough for increasing the number of agents. This factors could influence on the multimodal experience.

The measurement techniques that requires camera and screen monitors were performed with standard devices at 60 fps. It is ideal to have the highest frame rate possible to increase precision.

Finally, due to time constraints, there were not enough users for generalization of results, that is why the approach was a combination between objective data from the recordings and experience evaluation from a survey. Likewise, this constraints did not allow a deeper analysis of the recordings and survey to obtain more results and increase the findings regarding user experience.

6.4 Recommendations and Future Work

The findings of this work can be used for next iterations of the proposed system in order to improve user interaction and efficiency. Moreover, it can be inspirational for the design and implementation of other solutions and their corresponding evaluations.

If possible, some of the limitations mentioned in the previous section can be solved with better equipment, more users, and enough amount of time.

In terms of system efficiency, latency can be reduced by increasing the computational power and/or implementing the system in platforms that prioritize efficiency and multi-threading. Packet loss in the mixed reality device can be decreased if a more direct connection is possible, but taking care of user interaction.

Two aspects that the author considers of high relevance for future work are:

- The exploration of different strategies for autonomous behaviour and to what extent the agents can take control of the performance.
- A deeper evaluation of human factors (such as *enjoyment*) under music psychology frameworks.

6.5 Summary

This chapter interpreted and explained the results obtained in Chapter 5. The key findings in terms of system measurements and user interaction validates the proposed design for a multimodal human-machine music improvisation system.

For such system, the number of agents impacts on latency, computational resources, and quality of the sound output. This aspects has a low affectation for user interaction when the musical tempo is moderated, and in other cases human limitations does not perceive those affectations. In that sense, multimodality is essential for confirming senses for the aural, visual, and proprioceptive modes.

The results regarding user evaluation suggest that performers enjoy the system despite of having a feeling of losing the vision of the improvisation. Some of the participants forced this vision along the performance, and others try to follow the machine. As such, a future direction for this kind of systems is the exploration of strategies for musical agents involved in human-machine music interactions and their level of autonomy, as well as deeper studies in human factors under music psychology methods.

Chapter 7

Conclusion

This thesis presented the design, implementation, and evaluation of a human-machine music performance system under a multi-modal approach and based on autonomous agents. Its design is supported by a theoretical framework that includes *Musical Agents*, *Live Algorithms*, *Swarm Intelligence*, and *Musical XR* concepts.

The multimodal solution for user interaction relies on three cutting-edge technologies: *motion capture*, *spatial audio*, and *mixed reality*. The system integrates these technologies to a main core that affords musical improvisation and agents' interaction.

This system was implemented in Max/MSP/Jitter as one only unit that interconnects three subsystems that represent the technologies mentioned above, which are: an *optical motion capture system* (OptiTrack), an *8-channel loudspeakers array for spatial audio*, and a *mixed reality headset* (Microsoft HoloLens). The musical input is fed through a MIDI controller that works for a multi-track looper involved with the overall system. The music lines are synchronized through a global metronome.

For the evaluation of *efficiency*, several measurements were taken in terms of latency, as well as packet loss and other real-time values. Before this, the system was tested in order to find the amount of agents that the solution was able to tolerate so that the audio was not degraded. This number was 8. Thus, all the measurements were run for number of agents in a range from 1 to 8.

The most relevant findings from these measurements are the latency and jitter values in three aspects: between *pressing a key in the MIDI controller and the sound output*, *spatial audio placement*, and *sound to visualization*. For a system that works with 8 agents, this latency values are: 44.46 ms, 134.892 ms, and 6.749 ms. The jitter values are: 6.96 ms, - ms, and 13.44 ms. Note that the *spatial audio placement* does not have a jitter value, this is because it was measured with a low frame rate camera in the method defined, which means that its latency value has an error of +/- 16.67 ms.

Moreover, the packet loss in the communication between the system and the HoloLens is 23.89 %.

Additionally, 7 musicians participated in the evaluation of the system. The events generated from every performance and the audio output were recorded. They dealt with the values shown above, and despite of these limitations, they were able to improvise a musical piece for a considerable amount of time. The shortest performance lasted 24 minutes, and the longest one 43 minutes. They felt mostly that the machine changed their vision of the piece, which was undesirable for some users, while others tried to follow the machine. They agreed that the machine was attempting to replicate their style to some extent. As an overall evaluation of the experience, they rated the solution as *easy-to-use* and express a high *enjoyment*, which can be contrasted with the time they spent with the system.

The results obtained validates the way-of-making for the system, and the research question about *how to design and implement* such system is answered in Chapter 4 with the endorsement of the key findings described above. Therefore, it is possible the conception of the envisioned system as an integrated solution for human-machine musical performances.

The contributions of this work are: the design and implementation of the proposed system, the evaluation strategies, and the results of this study.

The limitations are related with: equipment (low resolution for spatial audio, limited field of view for HoloLens, PC no powerful enough for supporting more than 8 agents), measurement equipment (low frame rate cameras), and time restrictions (not enough participants for generalization, and there is still more data that can be analyzed). Despite of these limitations, the findings are relevant to validate the system and answer the research question.

It is recommendable to address the limitations described above, reduce latency and packet loss values, and search for more efficient implementations. For future work it is considered the exploration of other strategies for multi-agent behaviour and their level of participation in human-machine musical performances. Besides, a further evaluation of human aspects in the musicology field can be conducted by using this system.

References

Bartlette, C., Headlam, D., Bocko, M., & Velikic, G. (2006, 9). Effect of Network Latency on Interactive Musical Performance. *Music Perception*, *24*(1), 49–62. Retrieved from `https://online.ucpress.edu/mp/article/24/1/49/62315/Effect-of-Network-Latency-on-Interactive-Musical` doi: 10.1525/mp.2006.24.1.49

Blackwell, T. (2007). Swarming and Music. In *Evolutionary computer music* (pp. 194–217). London: Springer London. Retrieved from `http://link.springer.com/10.1007/978-1-84628-600-1_9` doi: 10.1007/978-1-84628-600-1{_}9

Blackwell, T., Bown, O., & Young, M. (2012). Live Algorithms: Towards Autonomous Computer Improvisers. In *Computers and creativity* (Vol. 9783642317, pp. 147–174). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from `http://link.springer.com/10.1007/978-3-642-31727-9_6` doi: 10.1007/978-3-642-31727-9{_}6

Bourguet, M.-L. (2003). Designing and Prototyping Multimodal Commands. *Human-Computer Interaction (INTERACT'03)*, 717–720.

Bown, O., Carey, B., & Eigenfeldt, A. (2015). Manifesto for a Musebot Ensemble: A platform for live interactive performance between multiple autonomous musical agents. *Interntaional Symposium on Electronic Art*(August).

Bullock, J., Michailidis, T., & Poyade, M. (2016). Towards a Live Interface for Direct Manipulation of Spatial Audio. *Icli*, 134–141.

Caschera, M. C., Ferri, F., & Grifoni, P. (2007). Multimodal interaction systems: information and time features. *International Journal of Web and Grid Services*, *3*(1), 82. Retrieved from `http://www.inderscience.com/link.php?id=12638` doi: 10.1504/IJWGS.2007.012638

Catak, M., AlRasheedi, S., AlAli, N., AlQallaf, G., AlMeri, M., & Ali, B. (2021, 9). Artificial Intelligence Composer. In *2021 international conference on innovation* *and intelligence for informatics, computing, and technologies (3ict)* (pp. 608–613). IEEE. Retrieved from `https://ieeexplore.ieee.org/document/9581896/` doi: 10.1109/3ICT53449.2021.9581896

Choi, I. (2018). Structured Reciprocity for Musical Performance with Swarm Agents as a Generative Mechanism. In *Advances in computer entertainment technology* (Vol. 10714, pp. 689–712). Cham: Springer International Publishing.

Claypool, K. T., & Claypool, M. (2007, 7). On frame rate and player performance in first person shooter games. *Multimedia Systems*, *13*(1), 3–17. Retrieved from `http://link.springer.com/10.1007/s00530-007-0081-1` doi: 10.1007/s00530-007-0081-1

Costagliola, M. (2018). *Multi-user shared augmented audio spaces using motion capture systems* (Vol. 2018-Augus).

Das, S., Glickman, S., Hsiao, F. Y., & Lee, B. (2017). Music Everywhere - Augmented Reality Piano Improvisation Learning System. *Proceedings of the 207 International Conference on New Interfaces for Musical Expression*, 511 - 512.

D'Ulizia, A. (2009). Exploring Multimodal Input Fusion Strategies. In *Multimodal human computer interaction and pervasive services* (pp. 34–57). IGI Global. Retrieved from `http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-60566-386-9.ch003` doi: 10.4018/978-1-60566-386-9.ch003

Gifford, T., Knotts, S., McCormack, J., Kalonaris, S., Yee-King, M., & D'Inverno, M. (2018). Computational systems for music improvisation. *Digital Creativity*, *29*(1), 19–36. Retrieved from `https://doi.org/10.1080/14626268.2018.1426613` doi: 10.1080/14626268.2018.1426613

Grani, F., Overholt, D., Erkut, C., Gelineck, S., Triantafyllidis, G., Nordahl, R., & Serafin, S. (2015). Spatial Sound and Multimodal Interaction in Immersive Environments. In *Proceedings of the audio mostly 2015 on interaction with sound - am '15* (Vol. 07-09-Octo, pp. 1–5). New York, New York, USA: ACM Press. Retrieved from `http://dl.acm.org/citation.cfm?doid=2814895.2814919` doi: 10.1145/2814895.2814919

Hamilton, R., Caceres, J.-P., Nanou, C., & Platz, C. (2011, 12). Multi-modal musical environments for mixed-reality performance. *Journal on Multimodal User In-*

terfaces, *4*(3-4), 147–156. Retrieved from http://link.springer.com/10.1007/s12193-011-0069-1 doi: 10.1007/s12193-011-0069-1

Haus, G., & Pollastri, E. (2000). A multimodal framework for music inputs (poster session). In *Proceedings of the eighth acm international conference on multimedia - multimedia '00* (pp. 382–384). New York, New York, USA: ACM Press. Retrieved from http://portal.acm.org/citation.cfm?doid=354384.354539 doi: 10.1145/354384.354539

Hermelin, B., O'Connor, N., Lee, S., & Treffert, D. (1989, 5). Intelligence and musical improvisation. *Psychological Medicine*, *19*(2), 447–457. Retrieved from https://www.cambridge.org/core/product/identifier/S0033291700012484/type/journal_article doi: 10.1017/S0033291700012484

Hockett, P., & Ingleby, T. (2016, 10). Augmented Reality with Hololens: Experiential Architectures Embedded in the Real World. (October). Retrieved from http://arxiv.org/abs/1610.04281 doi: 10.6084/m9.figshare.c.3470907

Lima, M. (2017). *The Book of Circles* (1st ed.). Princeton Architectural Press. Retrieved from https://papress.com/products/the-book-of-circles-visualizing-spheres-of-knowledge

Meng, Y., Kazeem, O., & Muller, J. C. (2007). A Swarm intelligence based coordination algorithm for distributed multi-agent systems. *2007 International Conference on Integration of Knowledge Intensive Multi-Agent Systems, KIMAS 2007*, 294–299. doi: 10.1109/KIMAS.2007.369825

Mills, A. W. (1958). On the Minimum Audible Angle. *Journal of the Acoustical Society of America*, *30*(4), 237–246. doi: 10.1121/1.1909553

Müller, J., Geier, M., Dicke, C., & Spors, S. (2014, 4). The BoomRoom: Mid-air Direct Interaction with Virtual Sound Sources. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 247–256). New York, NY, USA: ACM. Retrieved from https://dl.acm.org/doi/10.1145/2556288.2557000 doi: 10.1145/2556288.2557000

Murray-Rust, D., Smaill, A., & Edwards, M. (2006). MAMA: An architecture for interactive musical agents. *Frontiers in Artificial Intelligence and Applications*, *141*(Pais 2006), 36–40.

Nakagawa, R., Komatsubara, R., Ota, T., & Ohmura, H. (2018, 10). Air Maestros: : A Multi-User Audiovisual Experience Using MR. In *Proceedings of the symposium on spatial user interaction* (pp. 168–168). New York, NY, USA: ACM. Retrieved from `https://dl.acm.org/doi/10.1145/3267782.3274685` doi: 10.1145/3267782.3274685

O'Modhrain, S. (2011, 3). A Framework for the Evaluation of Digital Musical Instruments. *Computer Music Journal*, *35*(1), 28–42. Retrieved from `https://direct.mit.edu/comj/article/35/1/28-42/94317` doi: 10.1162/COMJ{_}a{_}00038

Riley, I. T. (2021). *Touching Light: A Framework for the Facilitation of Music-Making in Mixed Reality* (Doctoral dissertation, West Virginia University Libraries). doi: 10.33915/etd.8075

Robinson, F. A. (2020, 2). Audio Cells: A Spatial Audio Prototyping Environment for Human-Robot Interaction. In *Proceedings of the fourteenth international conference on tangible, embedded, and embodied interaction* (pp. 955–960). New York, NY, USA: ACM. Retrieved from `https://dl.acm.org/doi/10.1145/3374920.3374999` doi: 10.1145/3374920.3374999

Schuett, N. (2002). *The Effects of Latency on Ensemble Performance* (Doctoral dissertation, Stanford University). Retrieved from `https://ccrma.stanford.edu/groups/soundwire/publications/papers/schuett_honorThesis2002.pdf`

Selfridge, R., & Barthet, M. (2019). Augmented Live Music Performance using Mixed Reality and Emotion Feedback. In *Proceedings of the 10th international symposium on computer music multidisciplinary research cmmr* (pp. 210–221).

Solis, J., Petersen, K., & Takanishi, A. (2011). Interactive Musical System for Multimodal Musician-Humanoid Interaction. In *Springer tracts in advanced robotics* (Vol. 74, pp. 253–268). Retrieved from `http://link.springer.com/10.1007/978-3-642-22291-7_15` doi: 10.1007/978-3-642-22291-7{_}15

Stivers, T., & Sidnell, J. (2005, 1). Introduction: Multimodal interaction. *Semiotica*, *2005*(156), 1–20. Retrieved from `papers2://publication/uuid/DACB9737-5BF7-45A9-BA25-101FB0E59593https://www.degruyter.com/document/doi/10.1515/semi.2005.2005.156.1/html` doi: 10.1515/semi.2005.2005.156.1

Tan, Y., & Zheng, Z.-y. (2013, 3). Research Advance in Swarm Robotics. *Defence Technology*, *9*(1), 18–39. Retrieved from `http://dx.doi.org/10.1016/j.dt.2013.03.001https://linkinghub.elsevier.com/retrieve/pii/S221491471300024X` doi: 10.1016/j.dt.2013.03.001

Tatar, K., & Pasquier, P. (2019, 1). Musical agents: A typology and state of the art towards Musical Metacreation. *Journal of New Music Research*, *48*(1), 56–105. Retrieved from `https://www.tandfonline.com/doi/full/10.1080/09298215.2018.1511736` doi: 10.1080/09298215.2018.1511736

Tatar, K., Pasquier, P., & Siu, R. (2018, 4). REVIVE: An Audio-visual Performance with Musical and Visual AI Agents. In *Extended abstracts of the 2018 chi conference on human factors in computing systems* (Vol. 2018-April, pp. 1–6). New York, NY, USA: ACM. Retrieved from `https://dl.acm.org/doi/10.1145/3170427.3177771` doi: 10.1145/3170427.3177771

Tatar, K., Pasquier, P., & Siu, R. (2019). Audio-based Musical Artificial Intelligence and Audio-Reactive Visual Agents in Revive. In *Proceedings of the international computer music conference and new york city electroacoustic music festival.*

Thelle, N. J. W., & Pasquier, P. (2021). Spire Muse: A Virtual Musical Partner for Creative Brainstorming. In *Nime 2021.* PubPub. Retrieved from `https://nime.pubpub.org/pub/wcj8sjee` doi: 10.21428/92fbeb44.84c0b364

Turchet, L., Hamilton, R., & Camci, A. (2021). Music in Extended Realities. *IEEE Access*, *9*, 15810–15832. Retrieved from `https://ieeexplore.ieee.org/document/9328440/` doi: 10.1109/ACCESS.2021.3052931

Turner, K. (2009). *Balancing chorus and orchestra in performance: Problems and solutions for conductors of the nineteenth century and today* (Doctoral dissertation). doi: https://libres.uncg.edu/ir/uncg/listing.aspx?id=1946

Wessel, D., & Wright, M. (2002). Problems and prospects for intimate musical control of computers. *Computer Music Journal*, *26*(3), 11. doi: 10.1162/014892602320582945

Wulfhorst, R. D., Nakayama, L., & Vicari, R. M. (2003). A multiagent approach for musical interactive systems. In *Proceedings of the second international joint conference on autonomous agents and multiagent systems - aamas '03* (p. 584). New York, New York, USA: ACM Press. Retrieved from `http://portal.acm.org/citation.cfm?doid=860575.860669` doi: 10.1145/860575.860669

Zhang, Y., Agarwal, P., Bhatnagar, V., Balochian, S., & Zhang, X. (2014). Swarm intelligence and its applications. *The Scientific World Journal*, *2014*, 204294. Retrieved from `http://www.hindawi.com/journals/tswj/2014/204294/` doi: 10.1155/2014/204294

Appendix A

User Data Processing

A.1 User 0

A.1.1 The User and the Physical Space

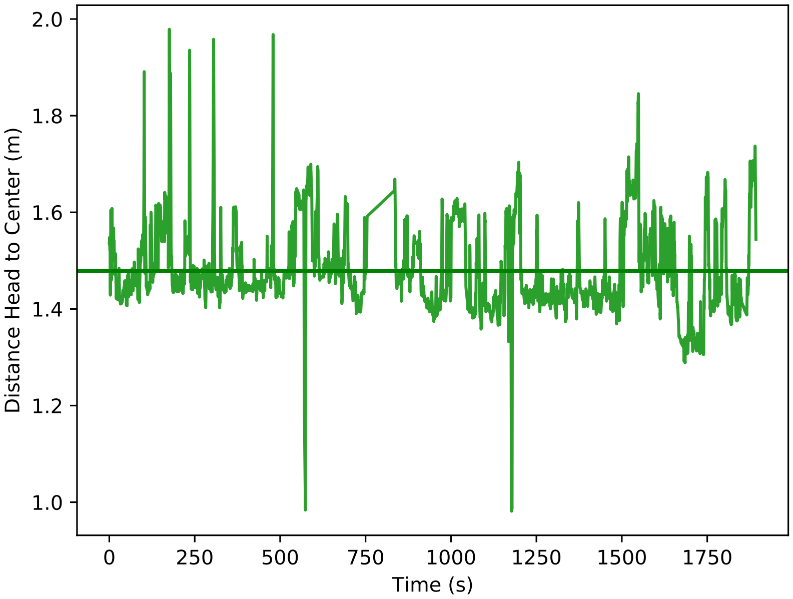


Figure A.1: Distance (in meters) from the head (HoloLens) to the center of the room during the performance session for *User 0*

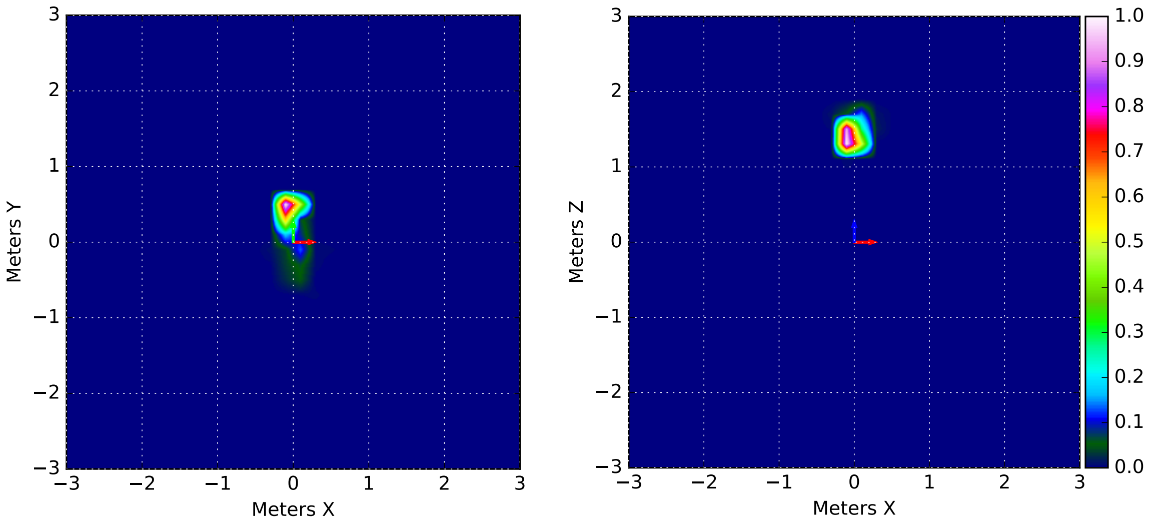


Figure A.2: Locations where *User 0* spent most of his or her time during the performance session. This view is from top (x, y) and from back (x, z).

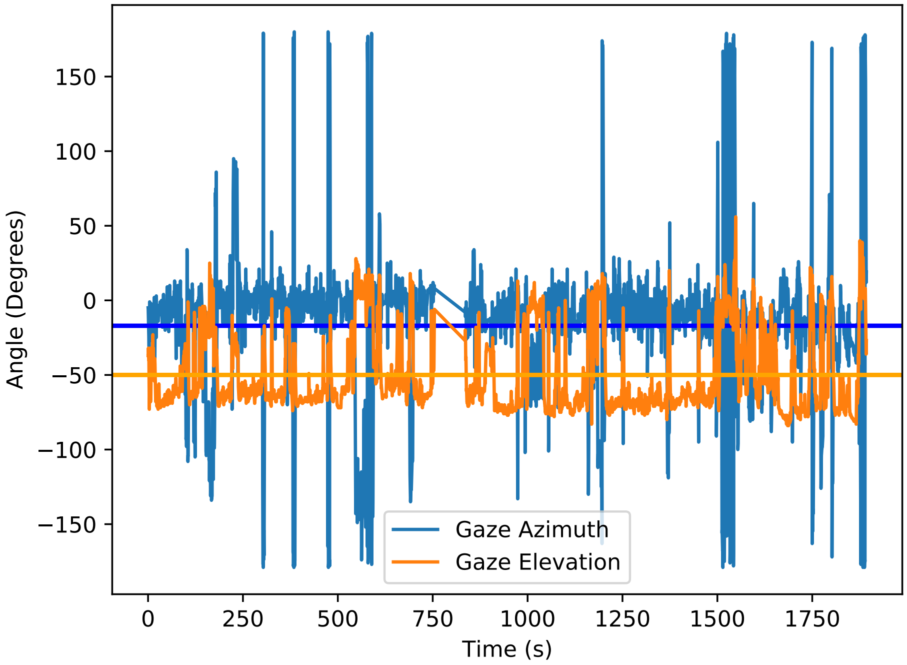


Figure A.3: Azimuth and Elevation angles (in degrees) for gaze direction regarding *User 0* during the performance session.

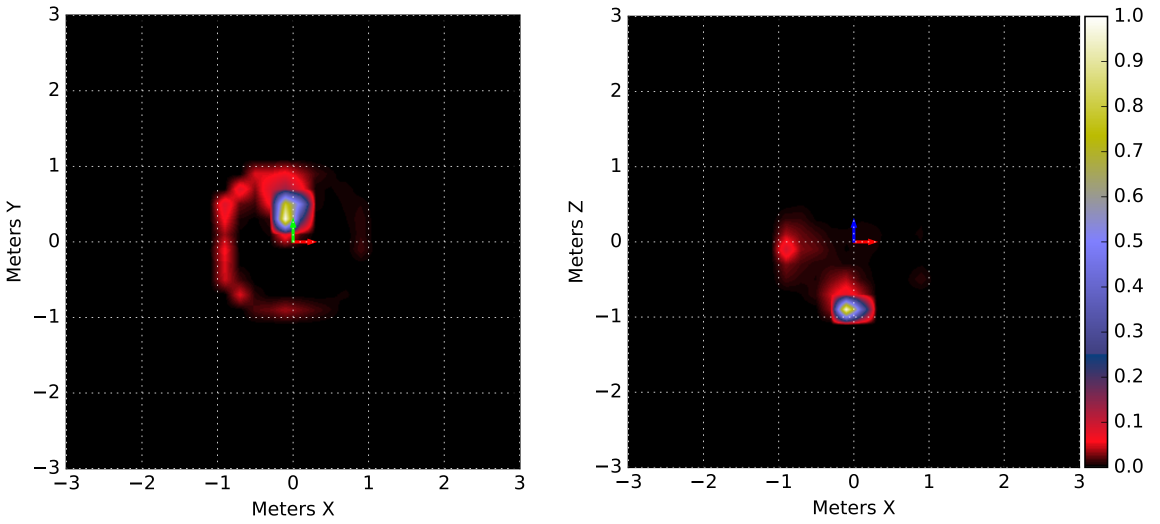


Figure A.4: Directions where *User 0* spent most of his or her time during the performance session. This view is from top (x, y) and from back (x, z). As this is a unit vector, the planes are just a reference and does not represent the actual room and its center.

A.1.2 The User and the Rigid Body

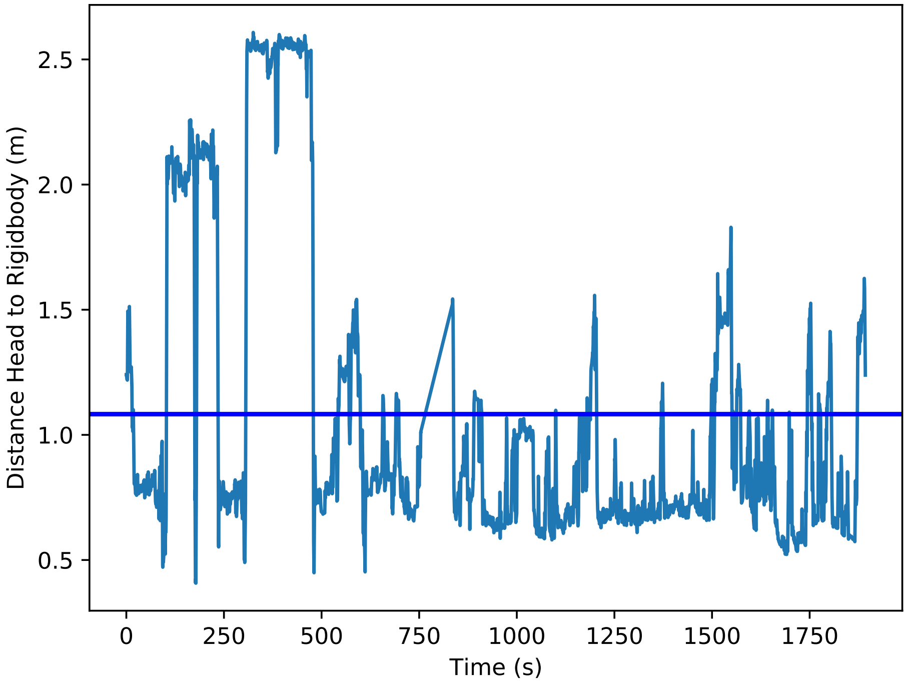


Figure A.5: Distance (in meters) from the head (HoloLens) to the rigid body during the performance session for *User 0*.

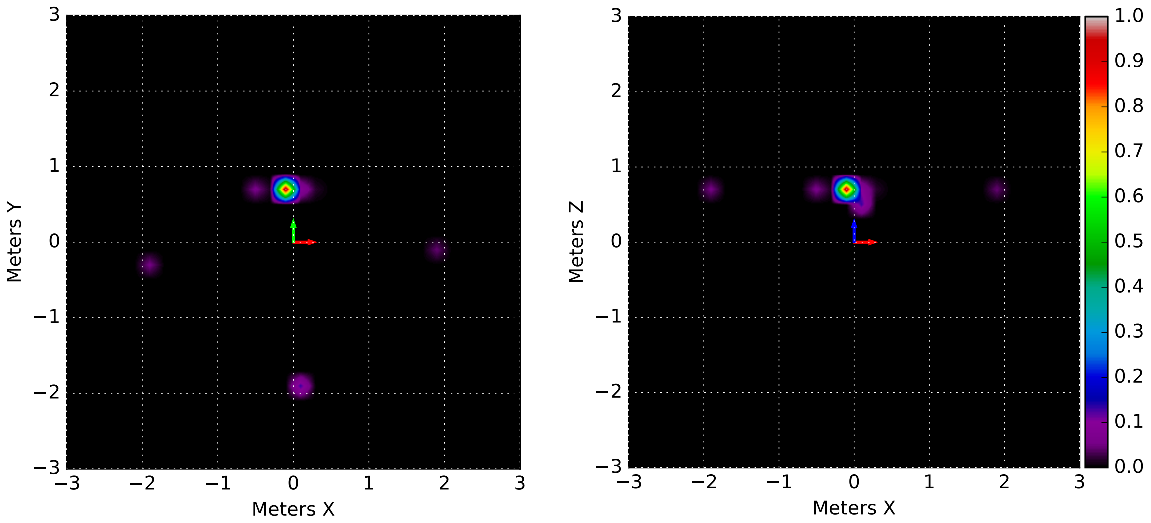


Figure A.6: Frequent locations where *User 0* placed the rigid body during the performance session. This view is from top (x, y) and from back (x, z).

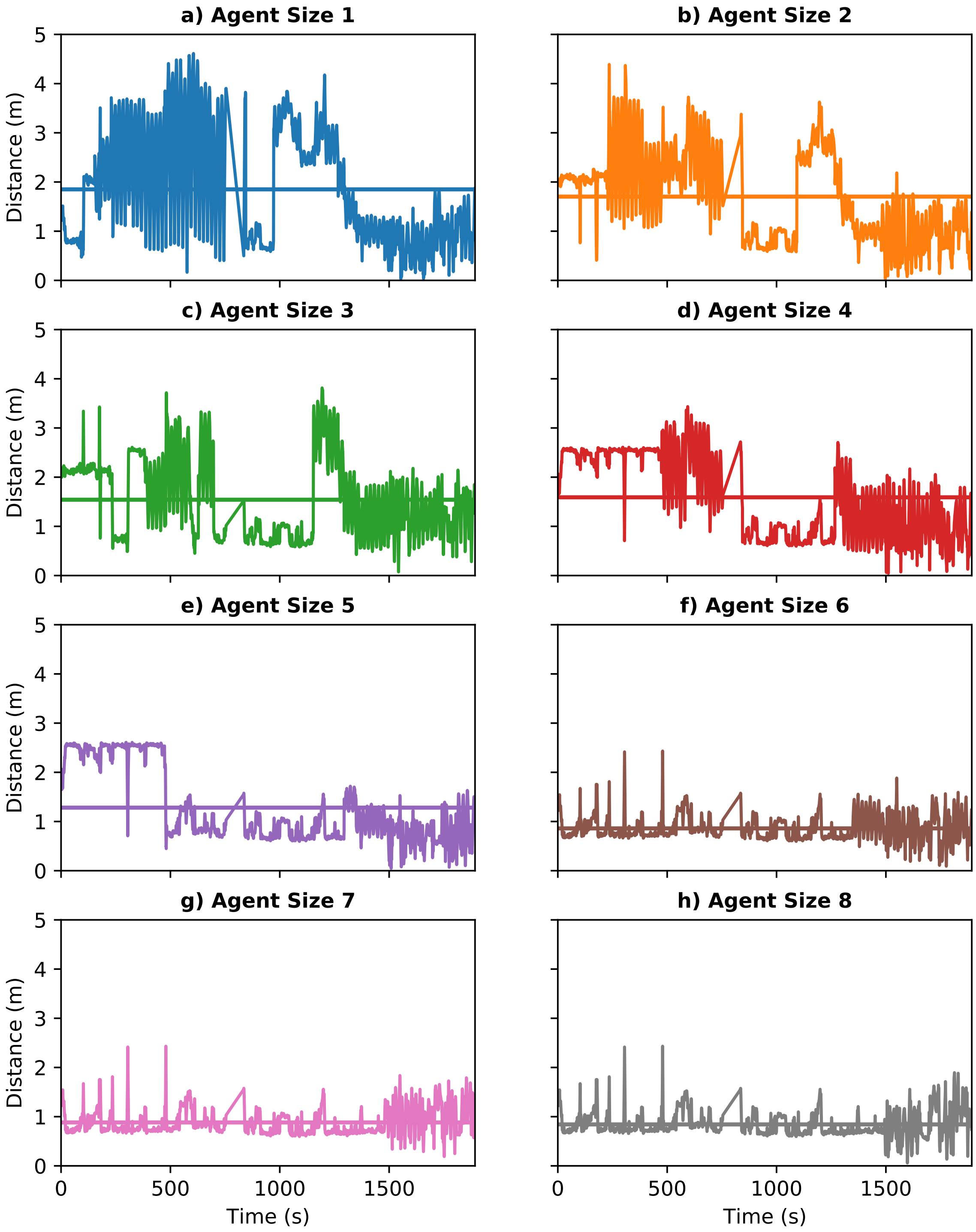


Figure A.7: Distance (in meters) from the head (HoloLens) to every agent during the performance session for *User 0*.

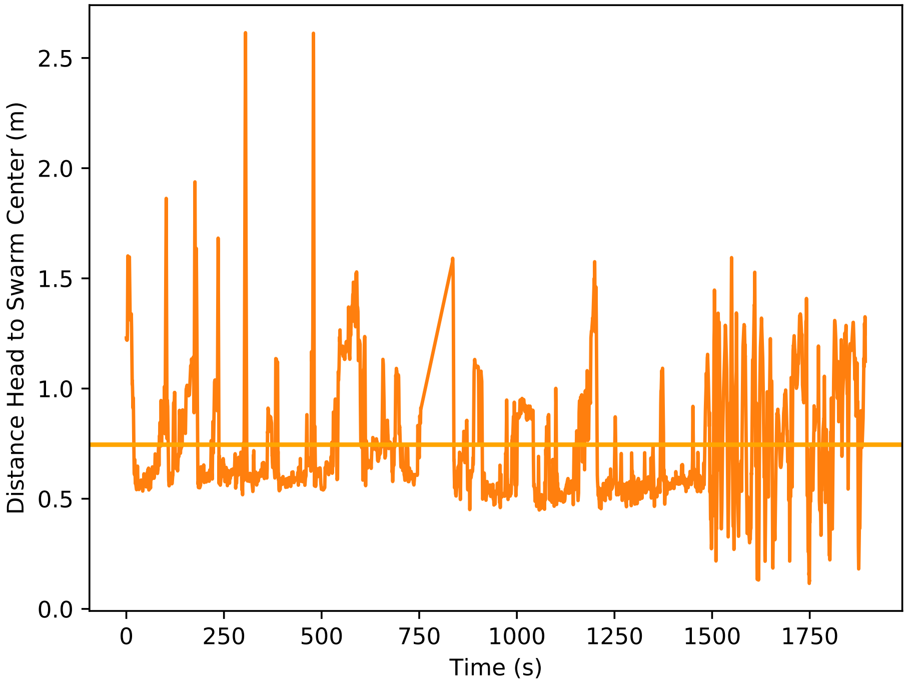


Figure A.8: Distance (in meters) from the head (HoloLens) to the agent's swarm center during the performance session for *User 0*.

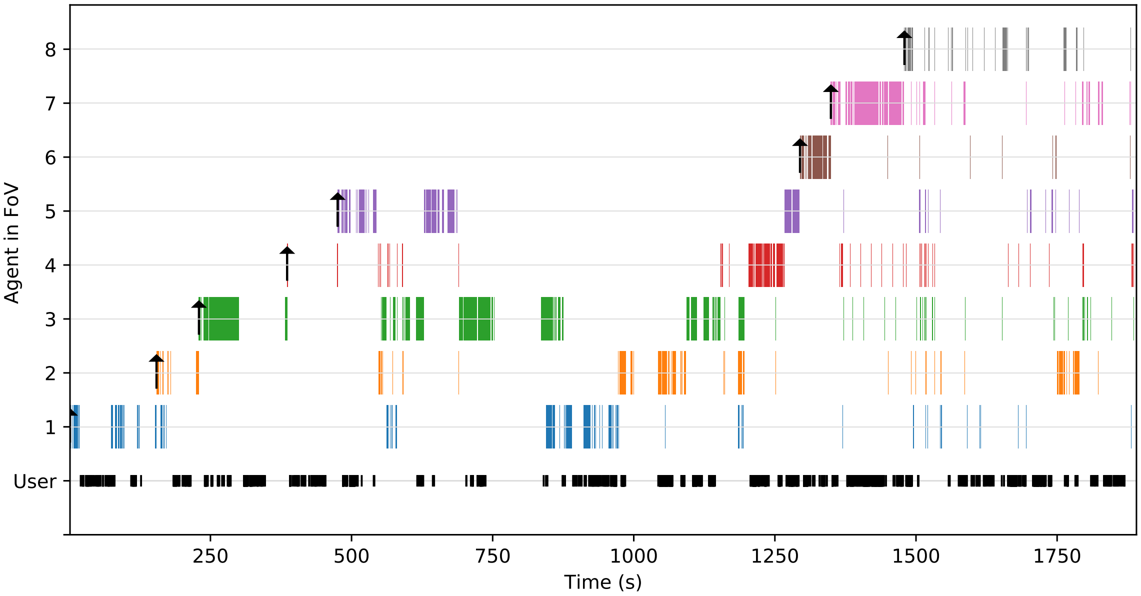


Figure A.9: Periods when agents were in the HoloLen's *field of view* (fov) during the performance session for *User 0*. The first row 'User' represents the user activity in the MIDI controller along the session. The arrows represent the moment of the first appearance of an agent.

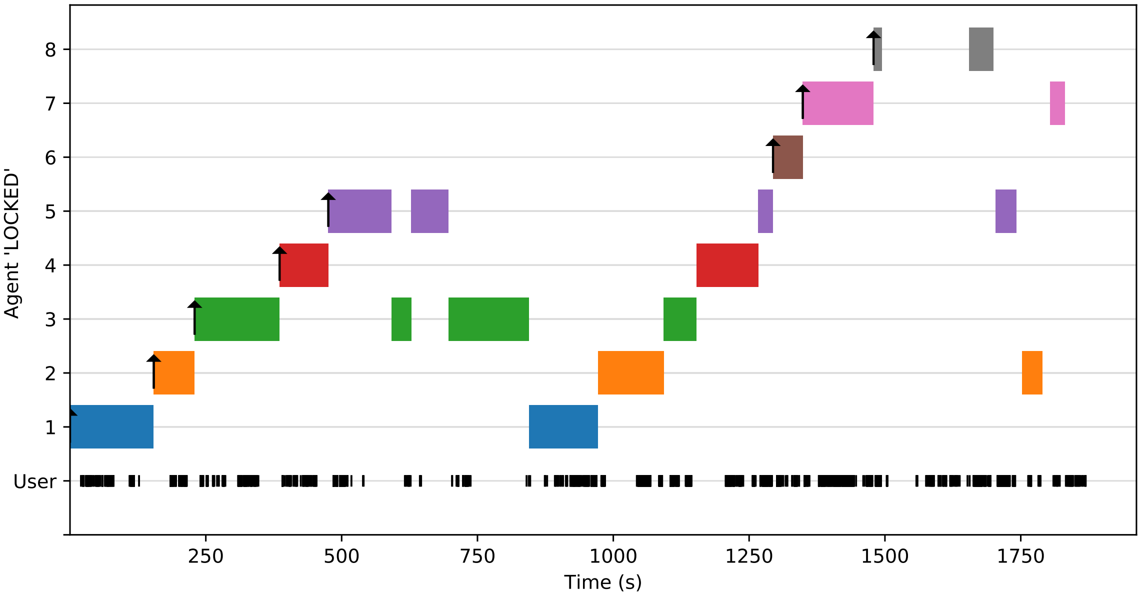


Figure A.10: Periods when agents were 'LOCKED' during the performance session for *User 0*. The first row 'User' represents the user activity in the MIDI controller along the session. The arrows represent the moment of the first appearance of an agent.

A.1.3 Agents Movement in a Session

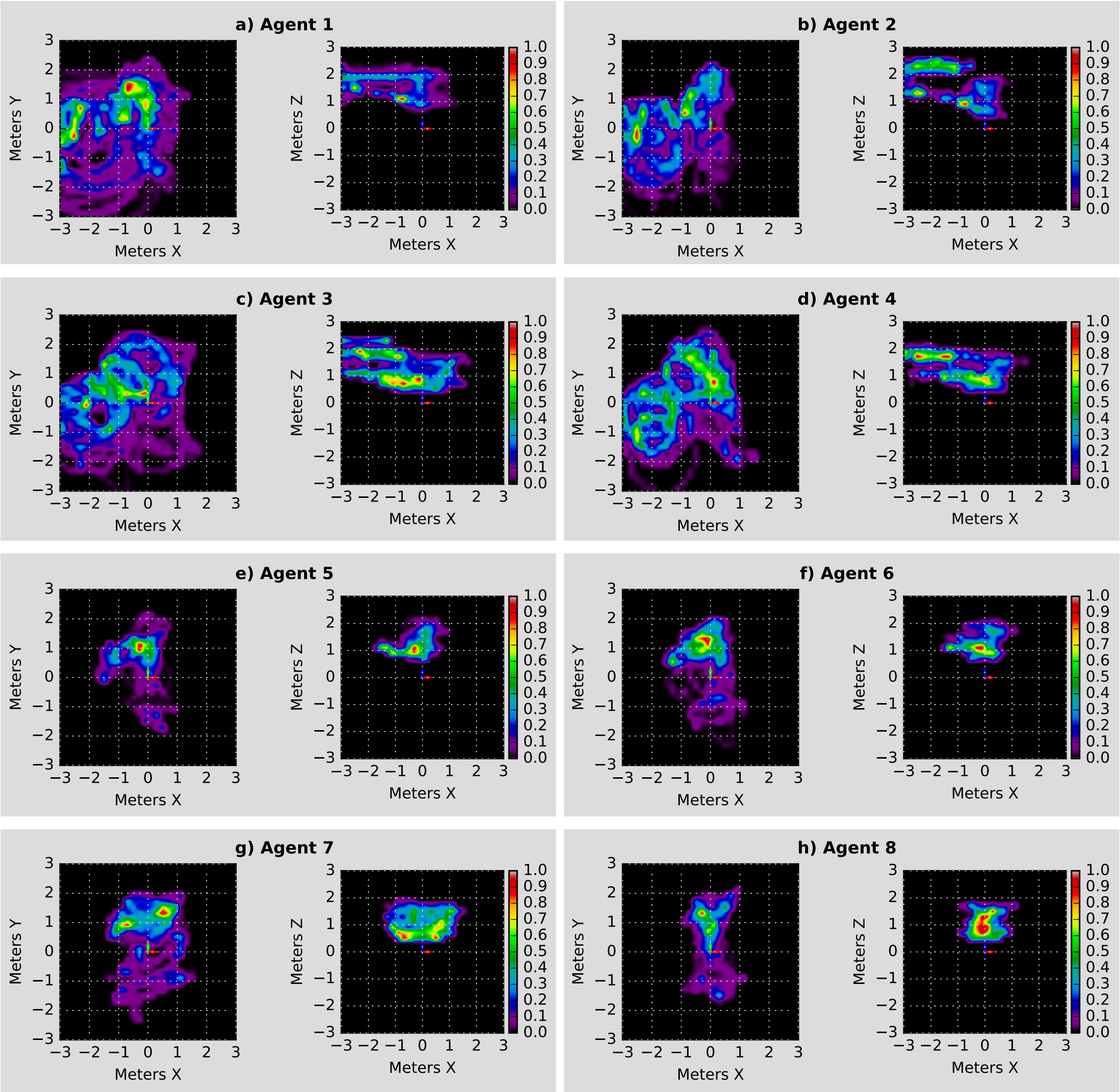


Figure A.11: Heatmaps for the movement of every agent during the performance session for *User 0*. It shows the locations where they spent most of their time, as well as trajectories. The view is from top (x, y) and from back (x, z) per each agent.

A.1.4 Tempo Changes

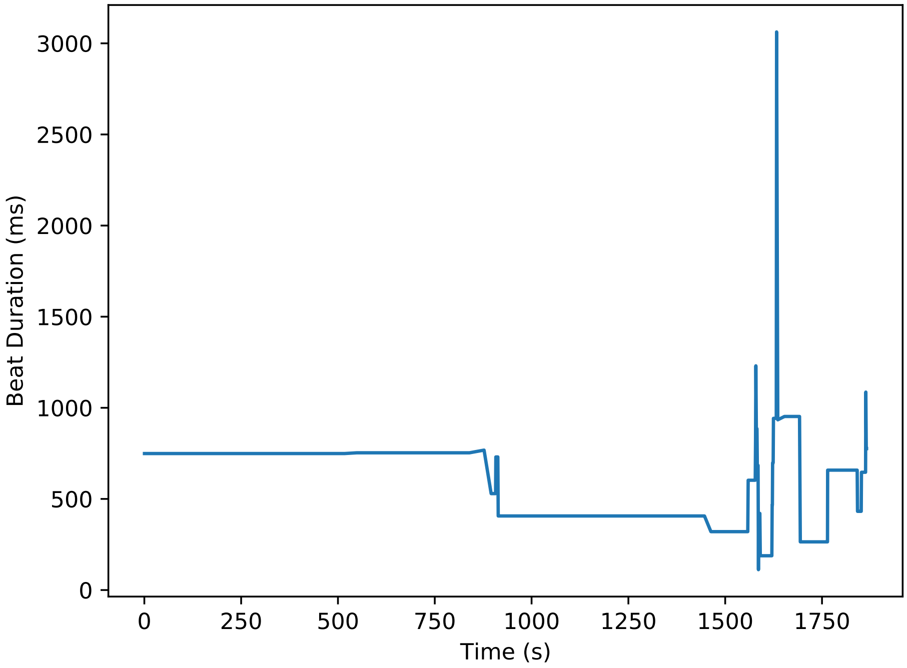


Figure A.12: Beat duration period for the metronome during the performance session for *User0*. This is limited to periods of 5000 ms to detect the actual BPM configuration given by the user at any point in time. The high impulses correspond to periods where the user changed the tempo or just stopped the metronome, wait for a moment, and played it again.

A.2 User 1

A.2.1 The User and the Physical Space

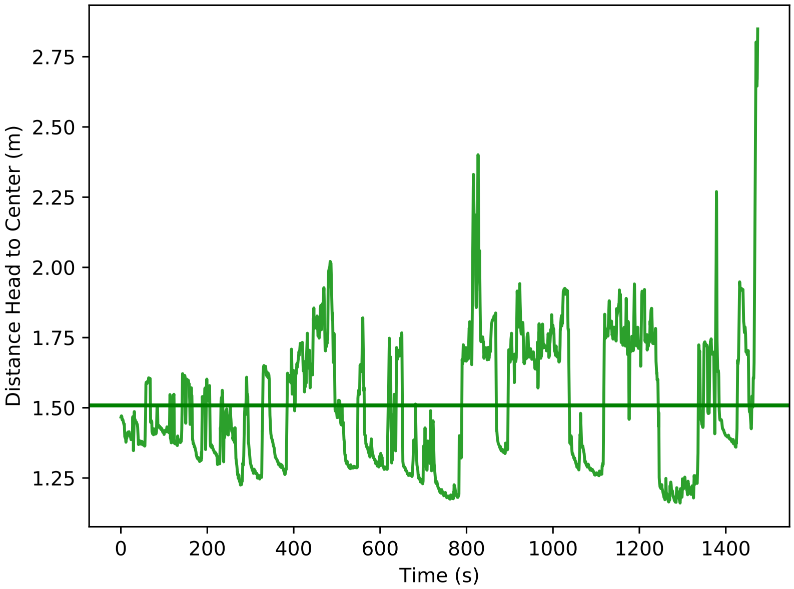


Figure A.13: Distance (in meters) from the head (HoloLens) to the center of the room during the performance session for *User 1*

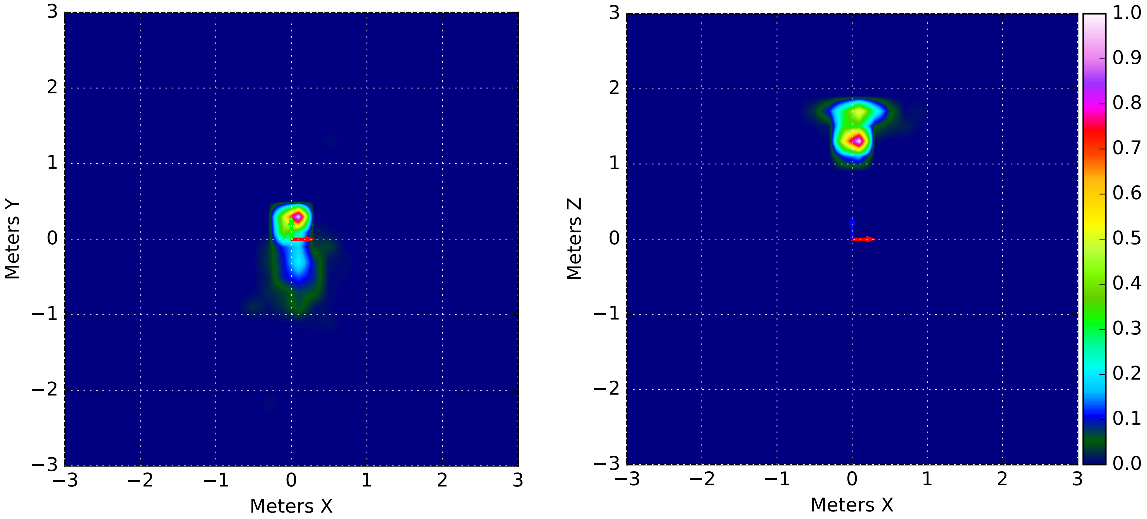


Figure A.14: Locations where *User 1* spent most of his or her time during the performance session. This view is from top (x, y) and from back (x, z).

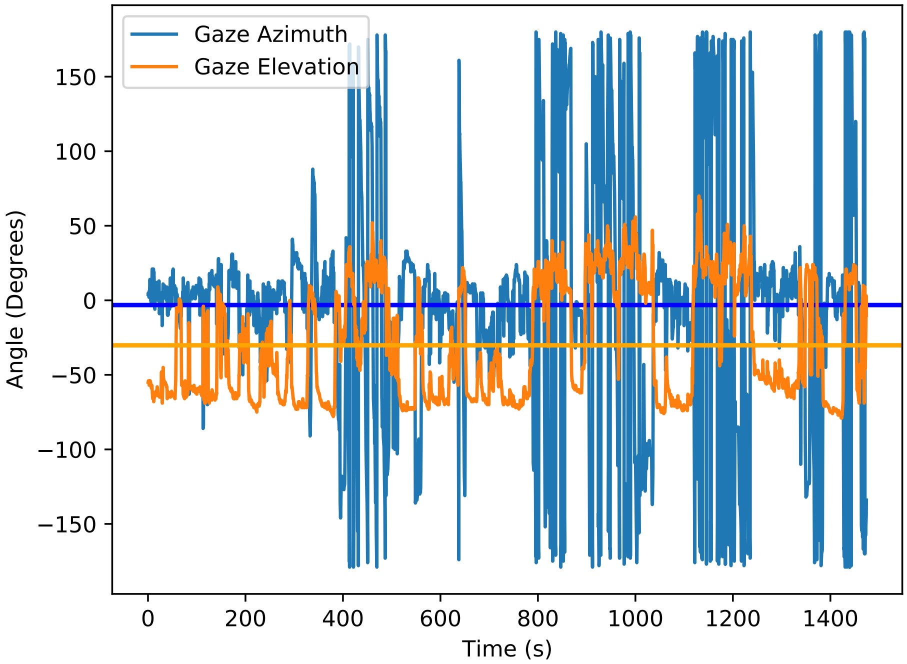


Figure A.15: Azimuth and Elevation angles (in degrees) for gaze direction regarding *User 1* during the performance session.

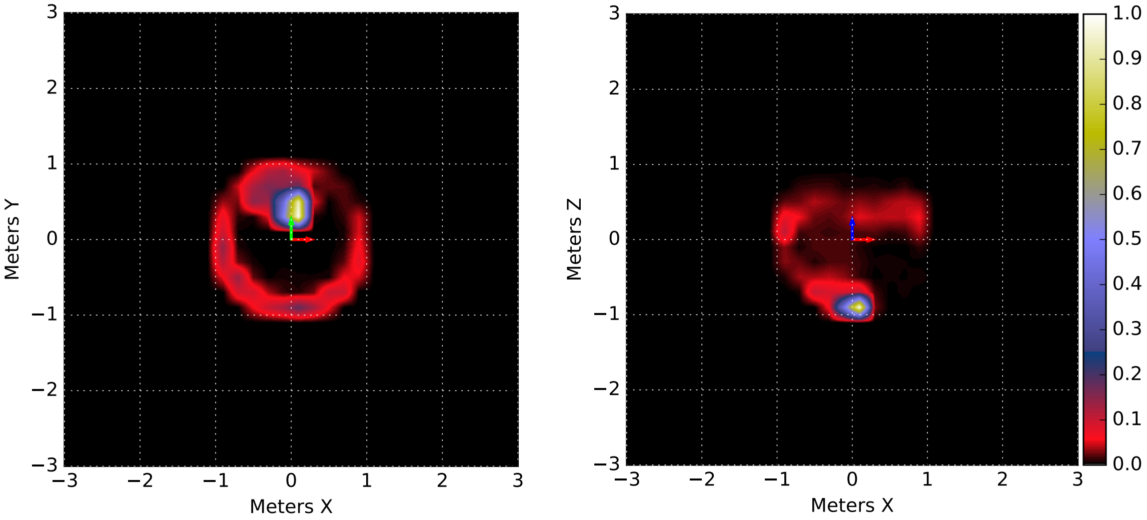


Figure A.16: Directions where *User 1* spent most of his or her time during the performance session. This view is from top (x, y) and from back (x, z). As this is a unit vector, the planes are just a reference and does not represent the actual room and its center.

A.2.2 The User and the Rigid Body

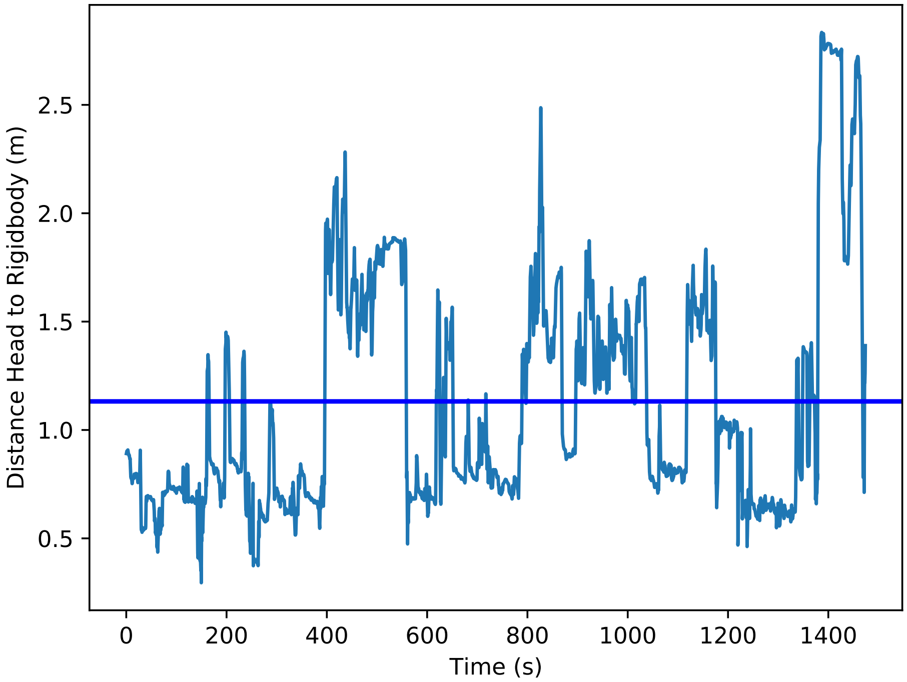


Figure A.17: Distance (in meters) from the head (HoloLens) to the rigid body during the performance session for *User 1*.

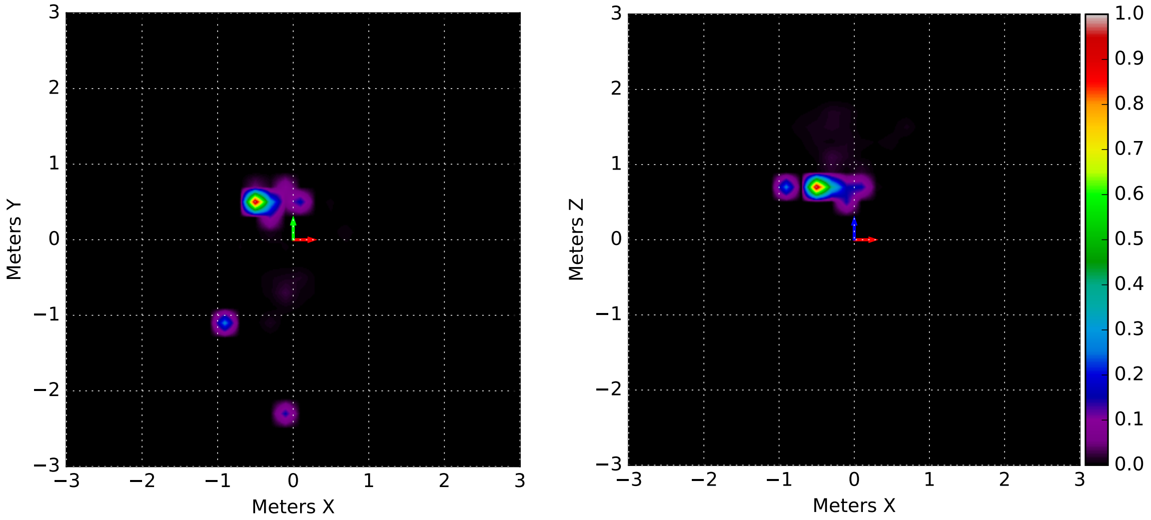


Figure A.18: Frequent locations where *User 1* placed the rigid body during the performance session. This view is from top (x, y) and from back (x, z).

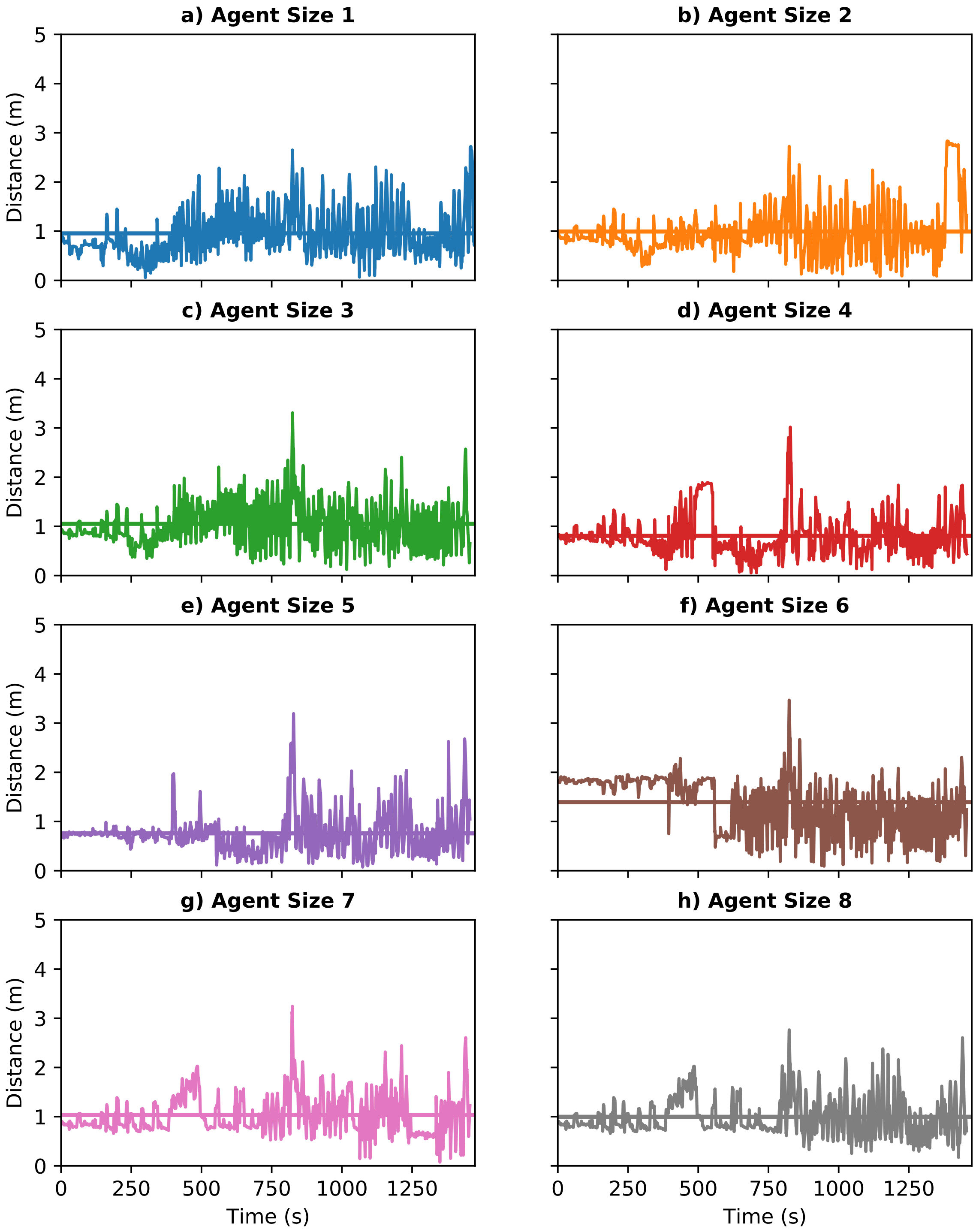


Figure A.19: Distance (in meters) from the head (HoloLens) to every agent during the performance session for *User 1*.

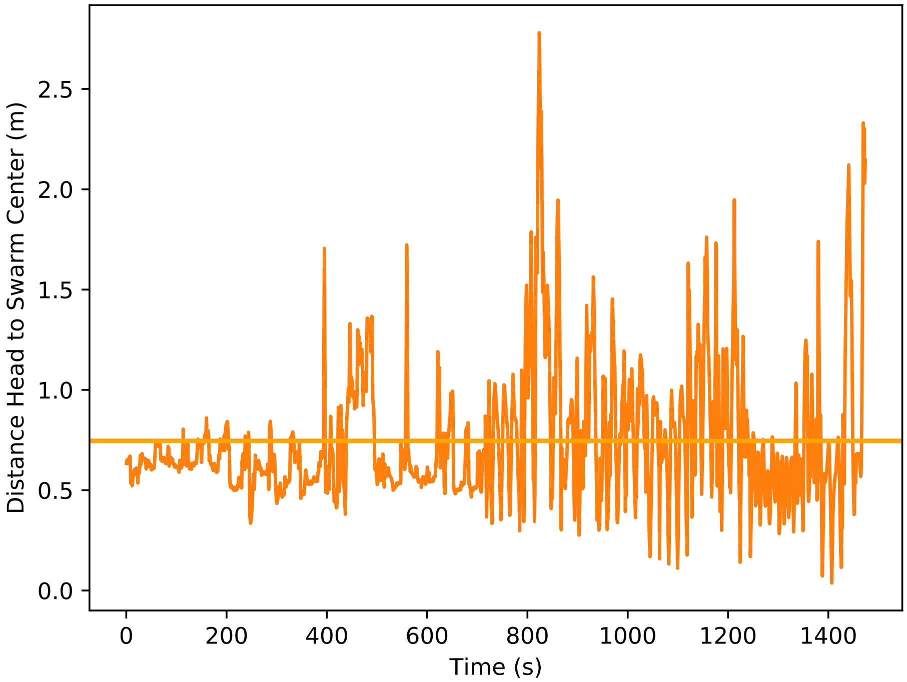


Figure A.20: Distance (in meters) from the head (HoloLens) to the agent's swarm center during the performance session for *User 1*.

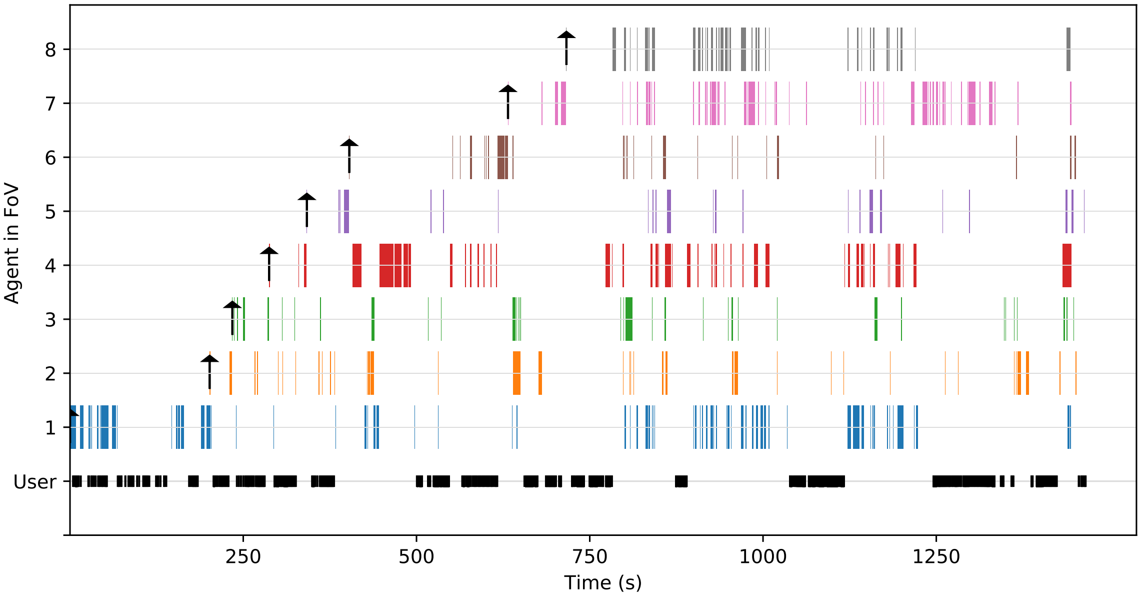


Figure A.21: Periods when agents were in the HoloLen's *field of view* (fov) during the performance session for *User 1*. The first row 'User' represents the user activity in the MIDI controller along the session. The arrows represent the moment of the first appearance of an agent.

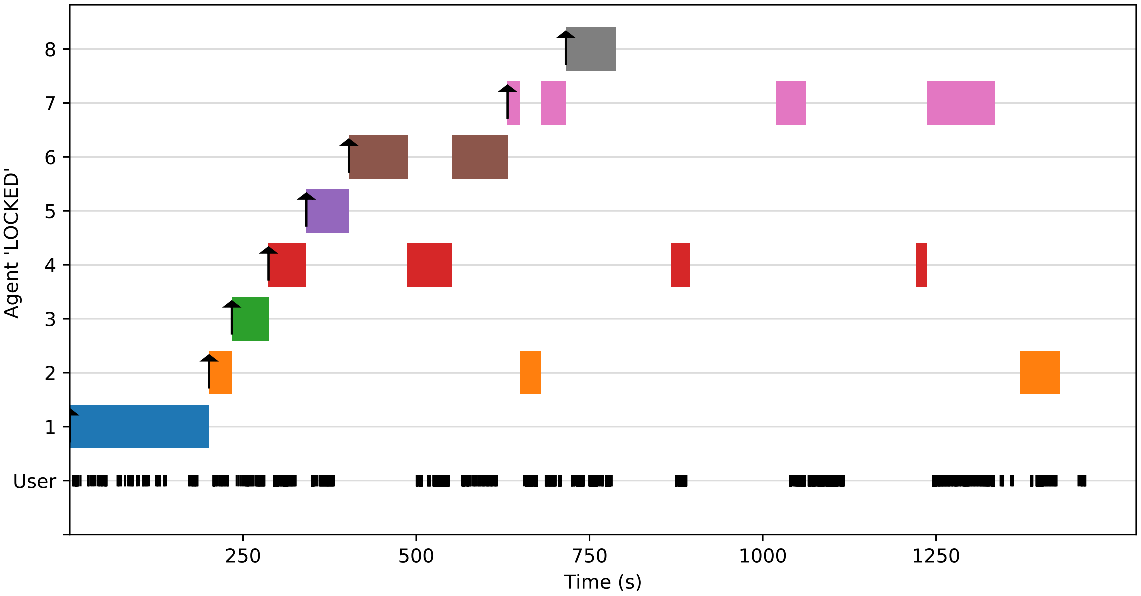


Figure A.22: Periods when agents were 'LOCKED' during the performance session for *User 1*. The first row 'User' represents the user activity in the MIDI controller along the session. The arrows represent the moment of the first appearance of an agent.

A.2.3 Agents Movement in a Session

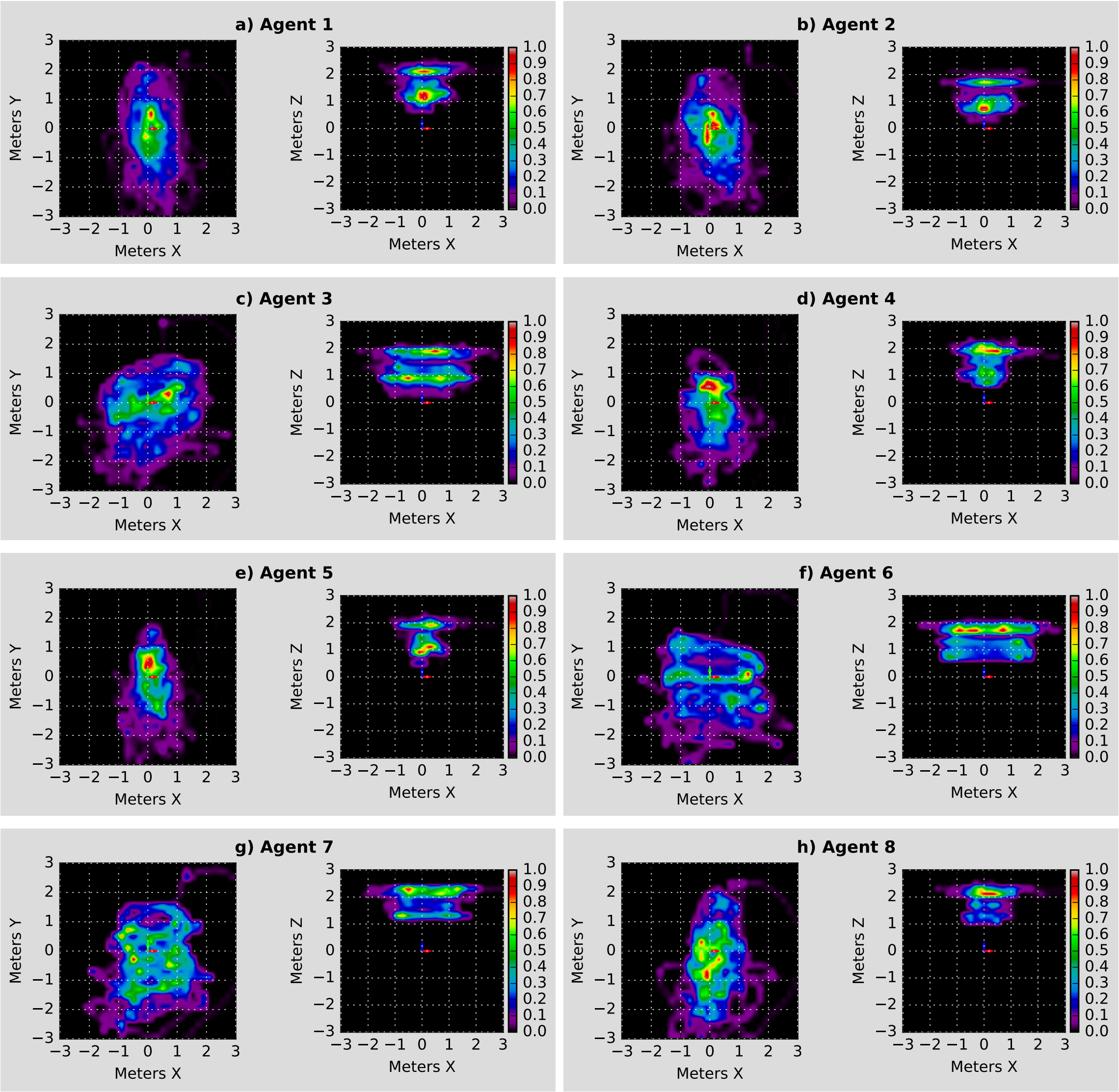


Figure A.23: Heatmaps for the movement of every agent during the performance session for *User 1*. It shows the locations where they spent most of their time, as well as trajectories. The view is from top (x, y) and from back (x, z) per each agent.

A.2.4 Tempo Changes

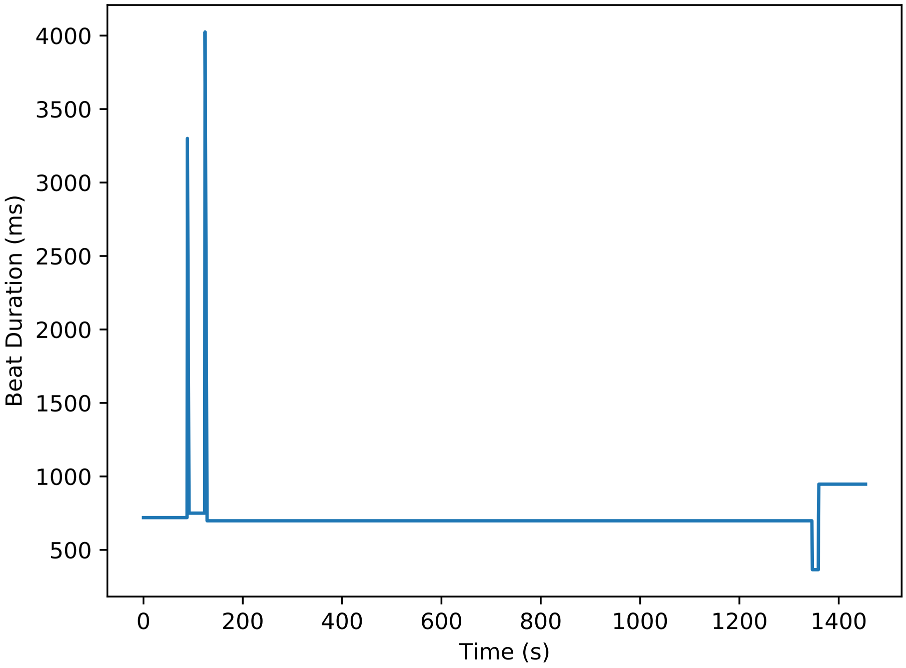


Figure A.24: Beat duration period for the metronome during the performance session for *User1*. This is limited to periods of 5000 ms to detect the actual BPM configuration given by the user at any point in time. The high impulses correspond to periods where the user changed the tempo or just stopped the metronome, wait for a moment, and played it again.

A.3 User 2

A.3.1 The User and the Physical Space

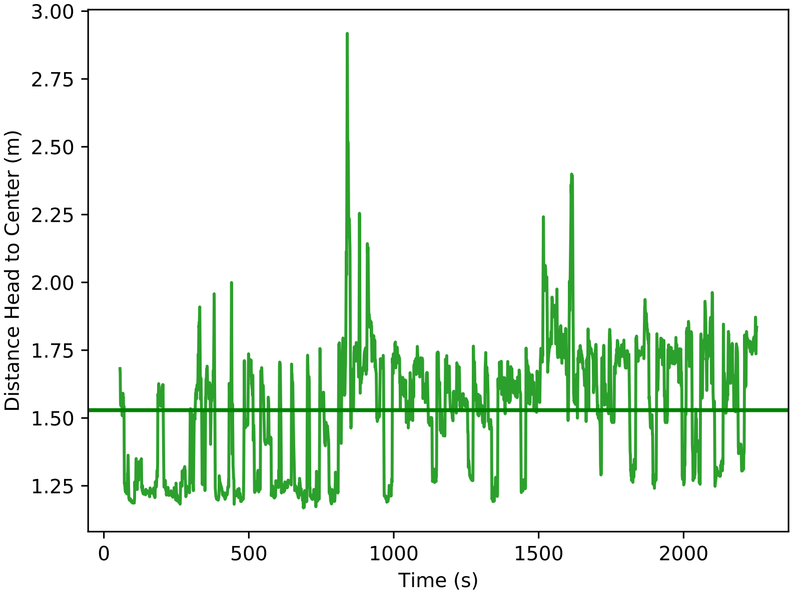


Figure A.25: Distance (in meters) from the head (HoloLens) to the center of the room during the performance session for *User 2*

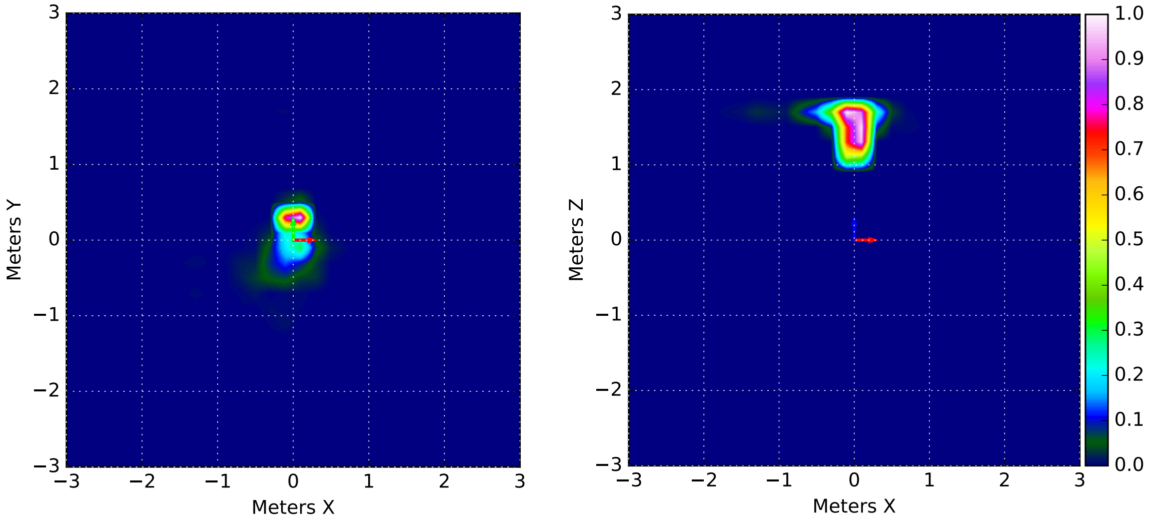


Figure A.26: Locations where *User 2* spent most of his or her time during the performance session. This view is from top (x, y) and from back (x, z).

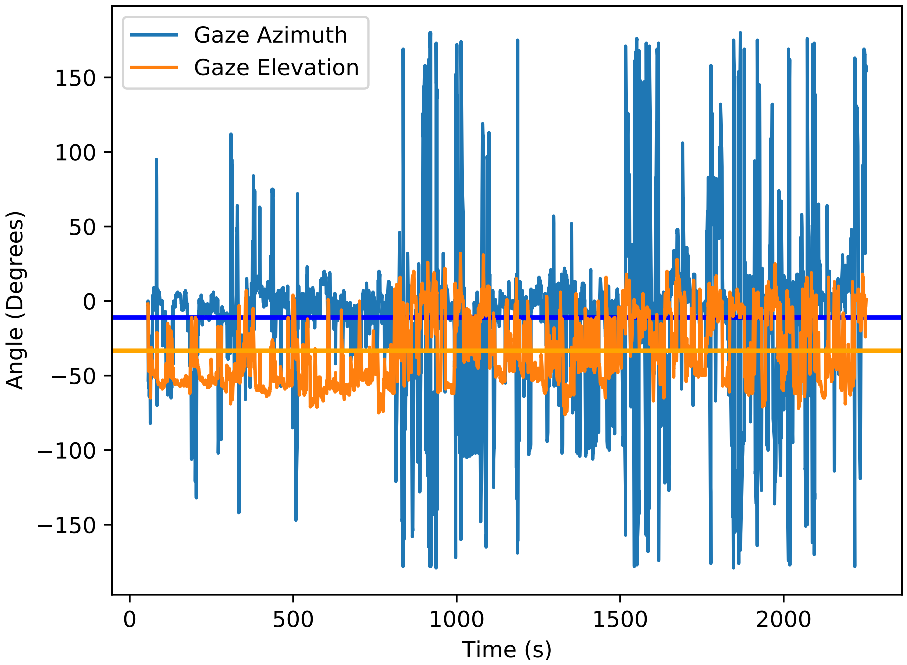


Figure A.27: Azimuth and Elevation angles (in degrees) for gaze direction regarding *User 2* during the performance session.

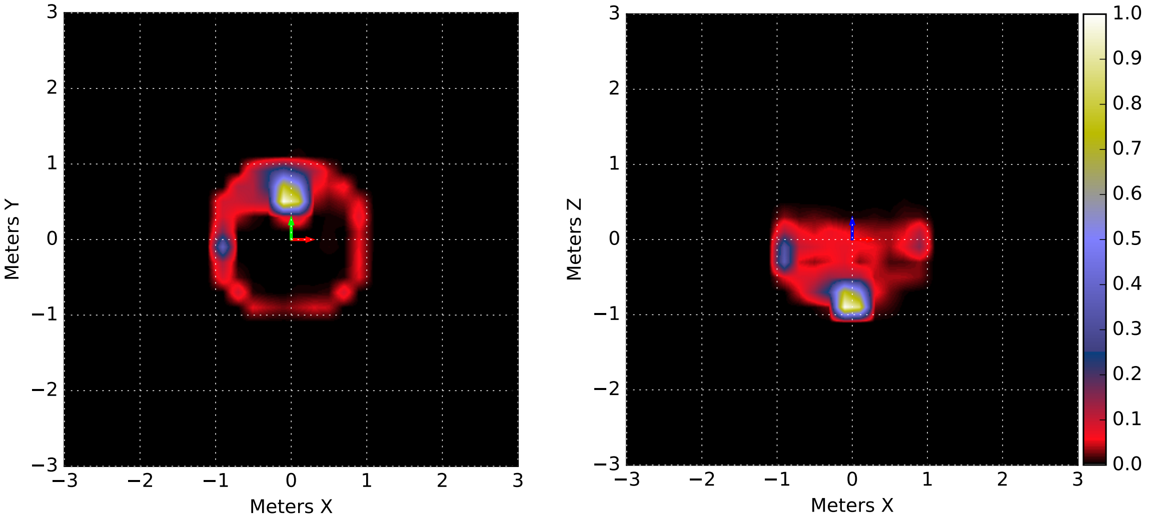


Figure A.28: Directions where *User 2* spent most of his or her time during the performance session. This view is from top (x, y) and from back (x, z). As this is a unit vector, the planes are just a reference and does not represent the actual room and its center.

A.3.2 The User and the Rigid Body

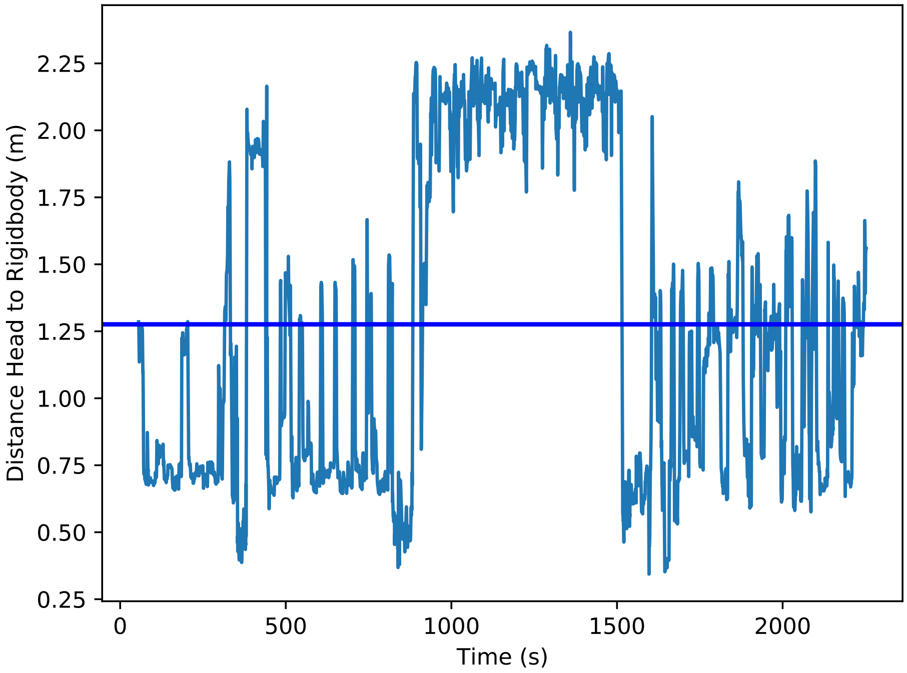


Figure A.29: Distance (in meters) from the head (HoloLens) to the rigid body during the performance session for *User 2*.

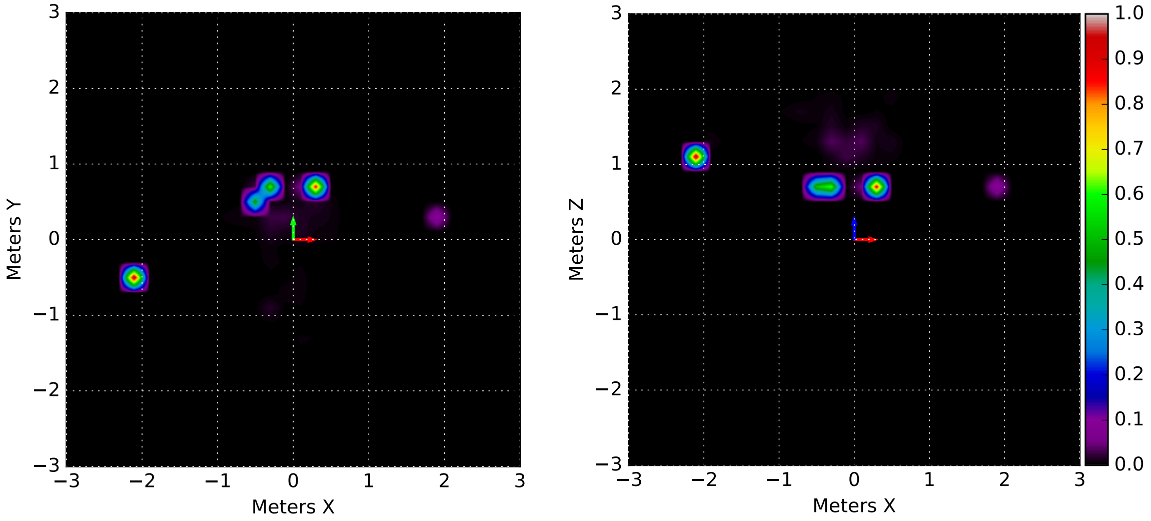


Figure A.30: Frequent locations where *User 2* placed the rigid body during the performance session. This view is from top (x, y) and from back (x, z).

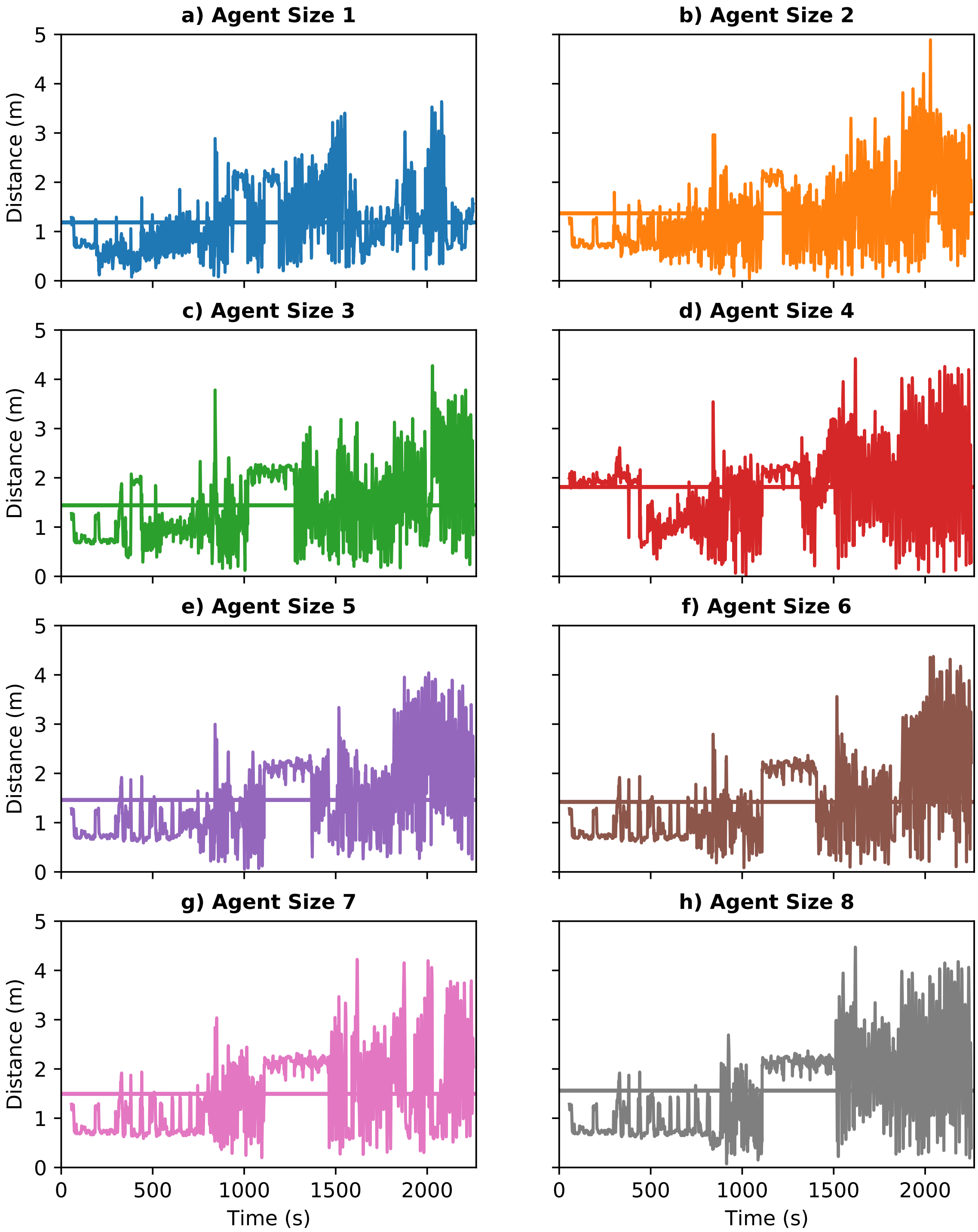


Figure A.31: Distance (in meters) from the head (HoloLens) to every agent during the performance session for *User 2*.

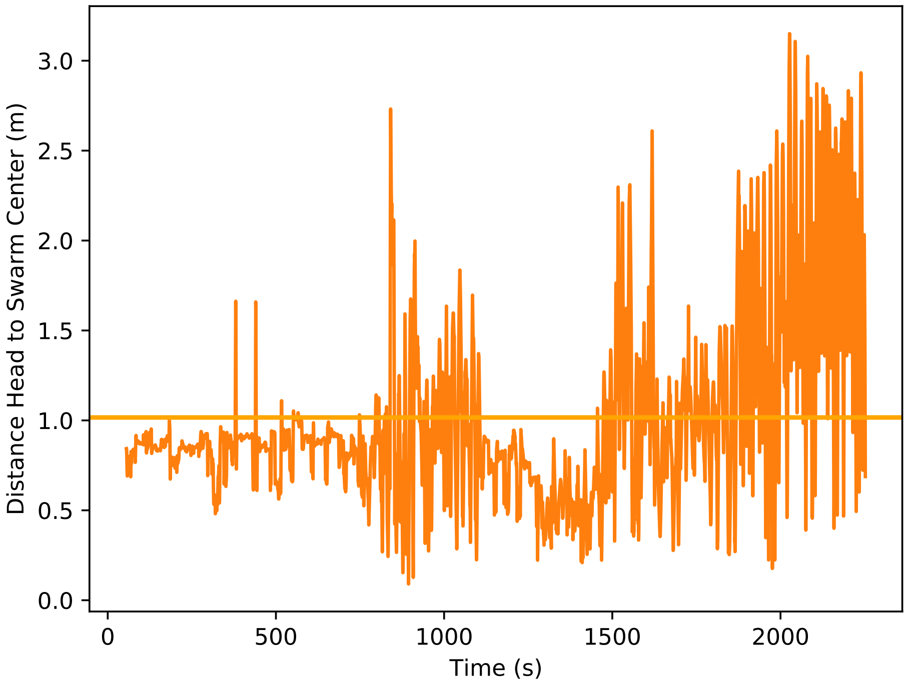


Figure A.32: Distance (in meters) from the head (HoloLens) to the agent's swarm center during the performance session for *User 2*.

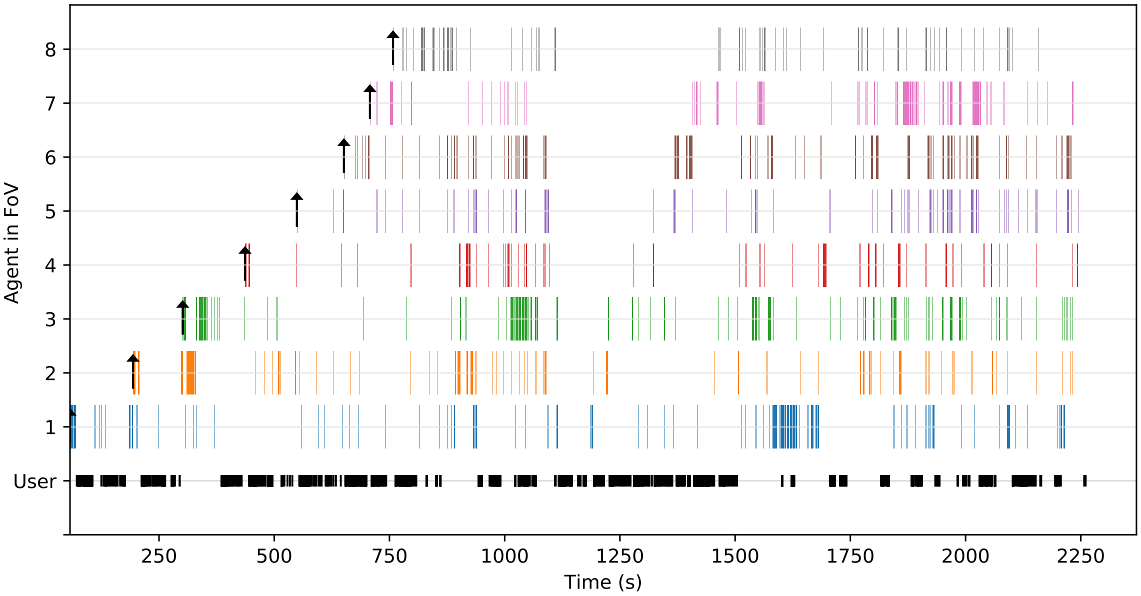


Figure A.33: Periods when agents were in the HoloLen's *field of view* (fov) during the performance session for *User 2*. The first row 'User' represents the user activity in the MIDI controller along the session. The arrows represent the moment of the first appearance of an agent.

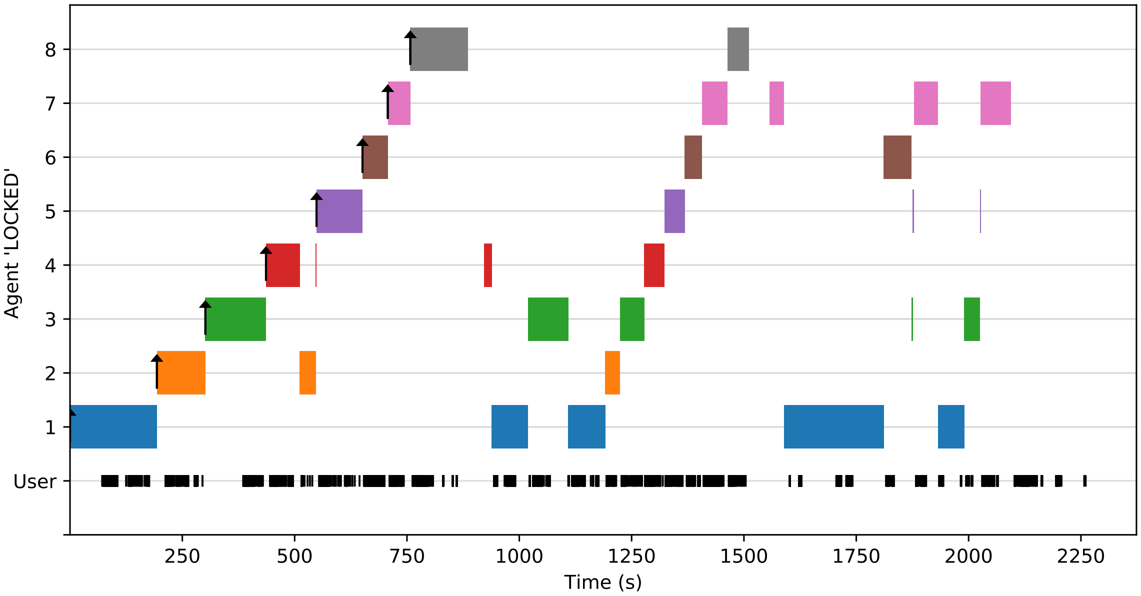


Figure A.34: Periods when agents were 'LOCKED' during the performance session for *User 2*. The first row 'User' represents the user activity in the MIDI controller along the session. The arrows represent the moment of the first appearance of an agent.

A.3.3 Agents Movement in a Session

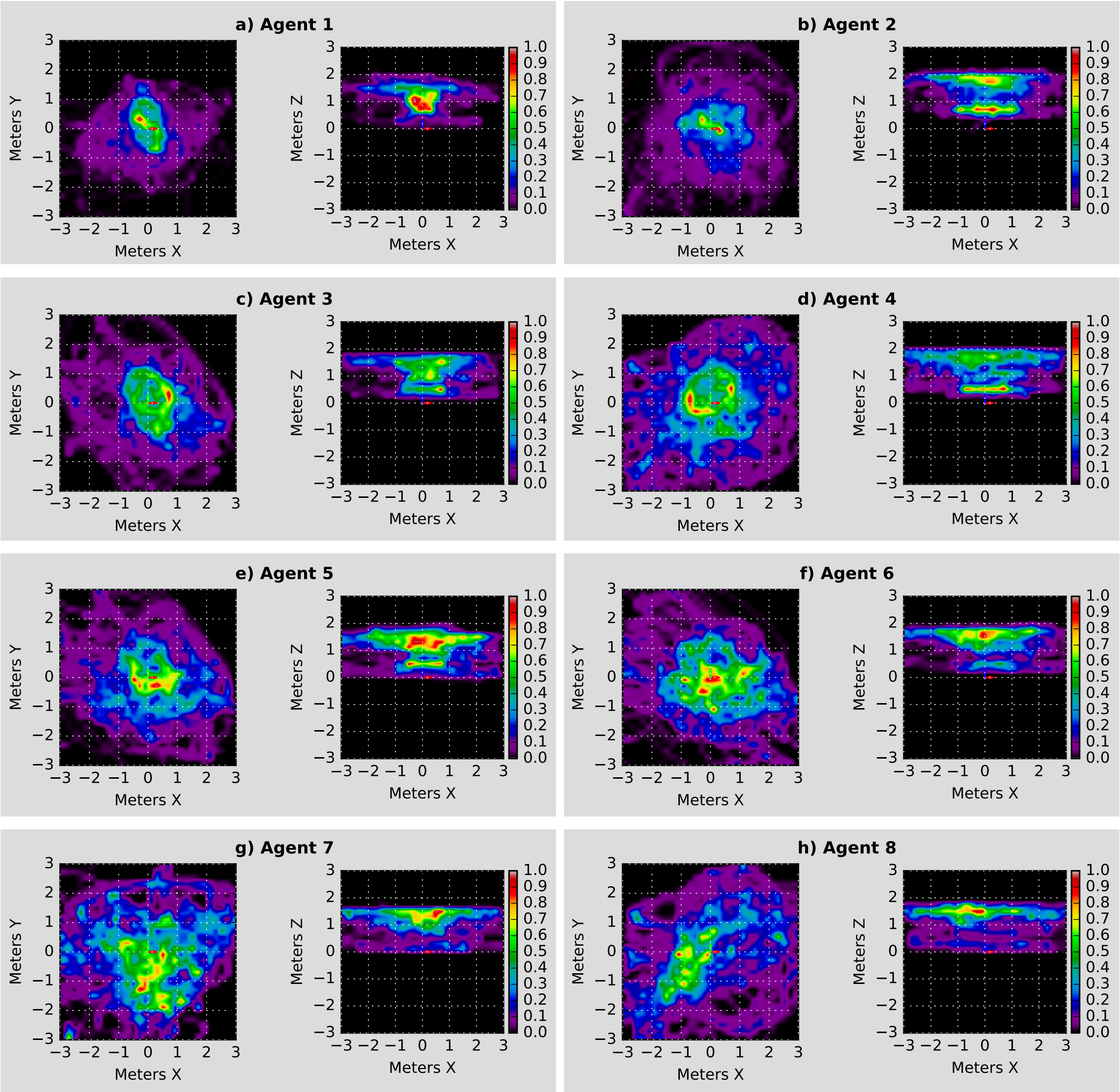


Figure A.35: Heatmaps for the movement of every agent during the performance session for *User 2*. It shows the locations where they spent most of their time, as well as trajectories. The view is from top (x, y) and from back (x, z) per each agent.

A.3.4 Tempo Changes

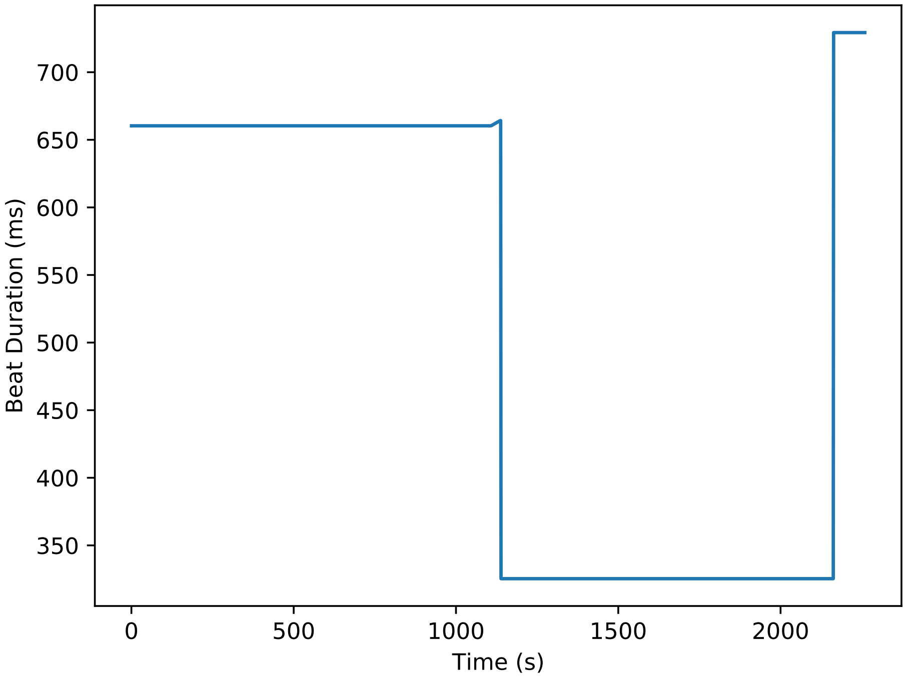


Figure A.36: Beat duration period for the metronome during the performance session for *User2*. This is limited to periods of 5000 ms to detect the actual BPM configuration given by the user at any point in time. The high impulses correspond to periods where the user changed the tempo or just stopped the metronome, wait for a moment, and played it again.

A.4 User 3

A.4.1 The User and the Physical Space

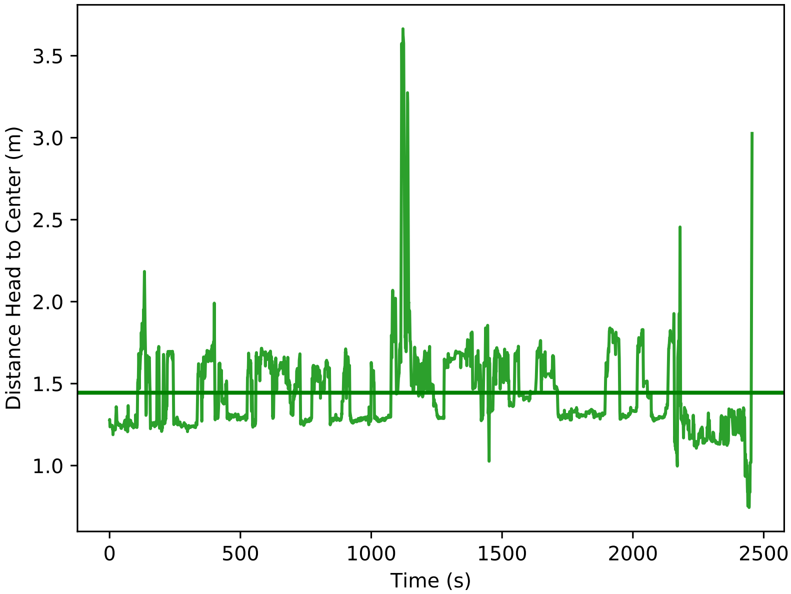


Figure A.37: Distance (in meters) from the head (HoloLens) to the center of the room during the performance session for *User 3*

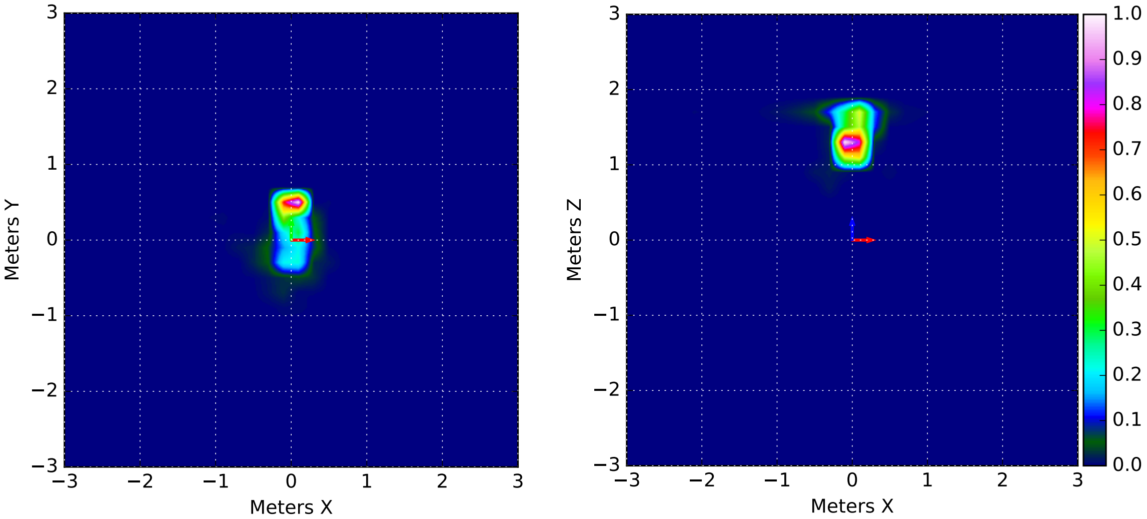


Figure A.38: Locations where *User 3* spent most of his or her time during the performance session. This view is from top (x, y) and from back (x, z).

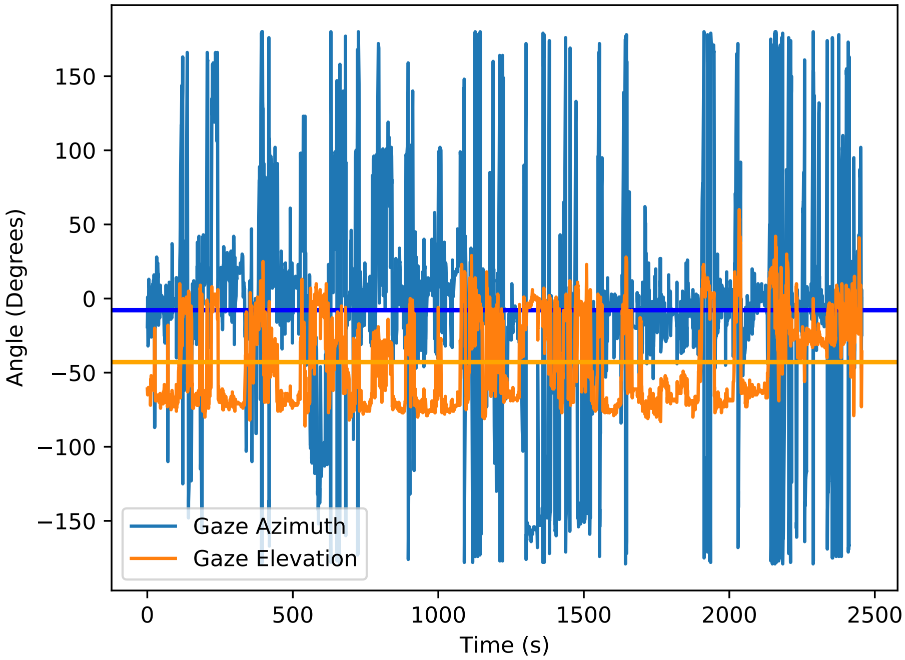


Figure A.39: Azimuth and Elevation angles (in degrees) for gaze direction regarding *User 3* during the performance session.

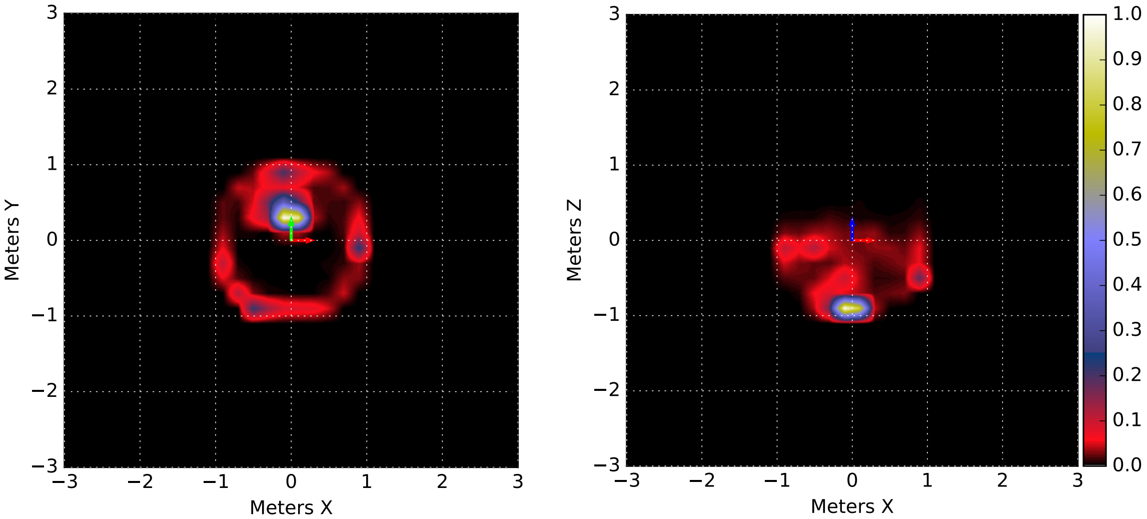


Figure A.40: Directions where *User 3* spent most of his or her time during the performance session. This view is from top (x, y) and from back (x, z). As this is a unit vector, the planes are just a reference and does not represent the actual room and its center.

A.4.2 The User and the Rigid Body

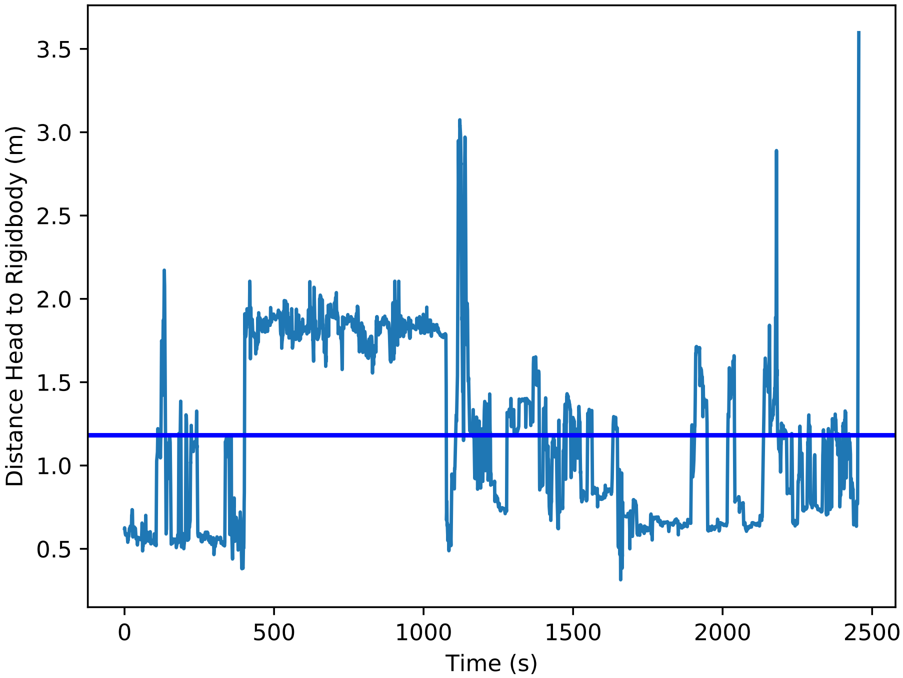


Figure A.41: Distance (in meters) from the head (HoloLens) to the rigid body during the performance session for *User 3*.

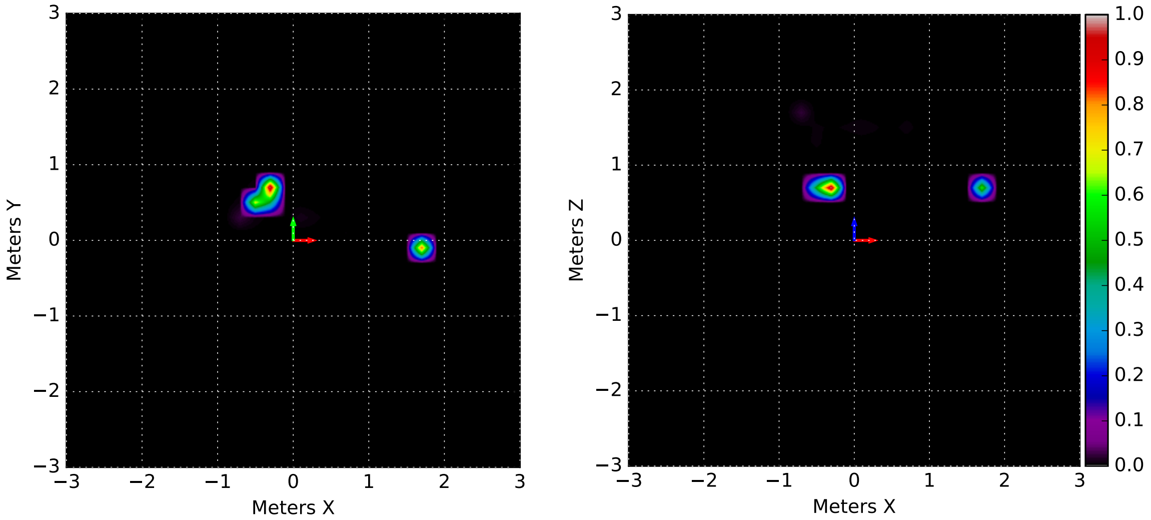


Figure A.42: Frequent locations where *User 3* placed the rigid body during the performance session. This view is from top (x, y) and from back (x, z).

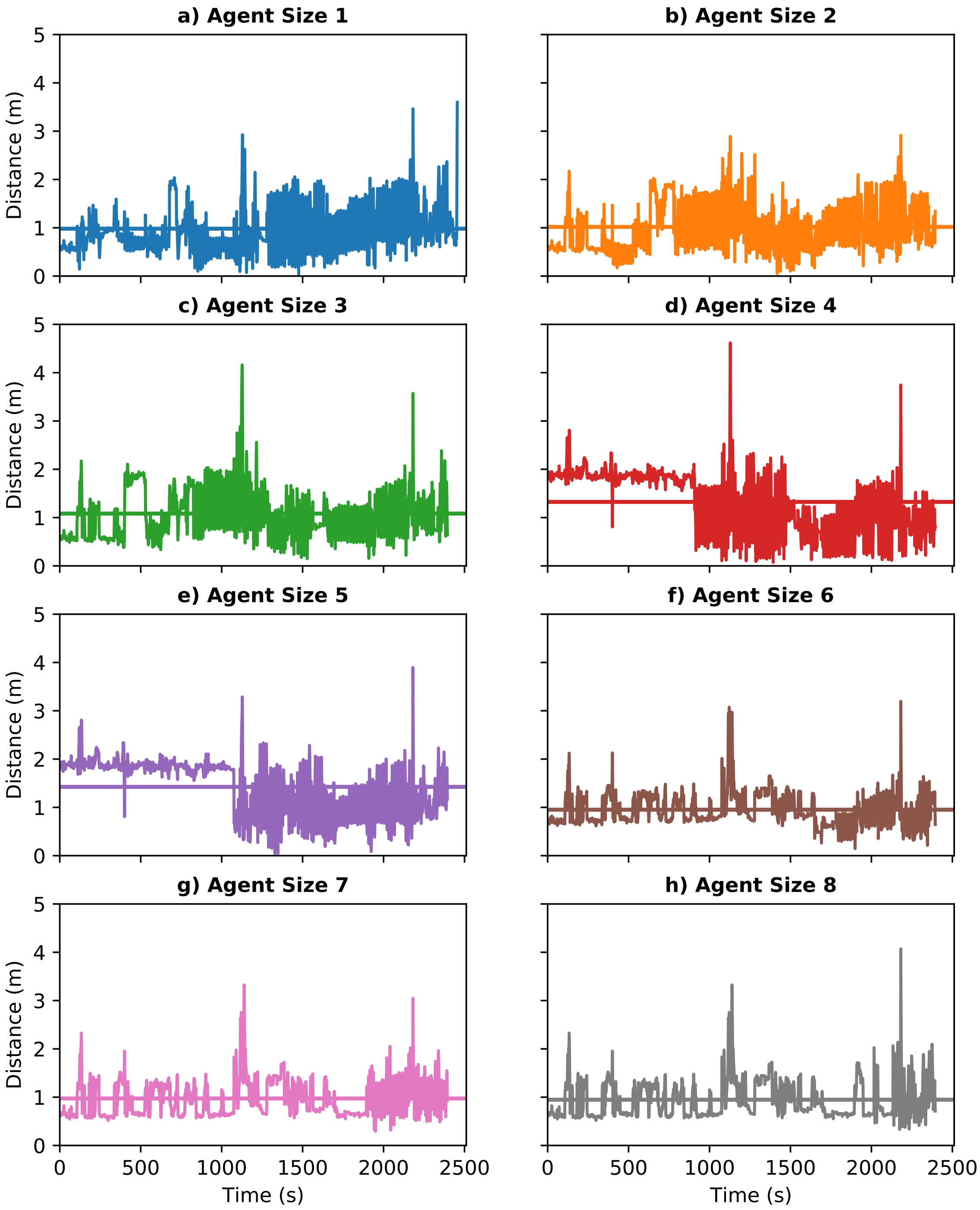


Figure A.43: Distance (in meters) from the head (HoloLens) to every agent during the performance session for *User 3*.

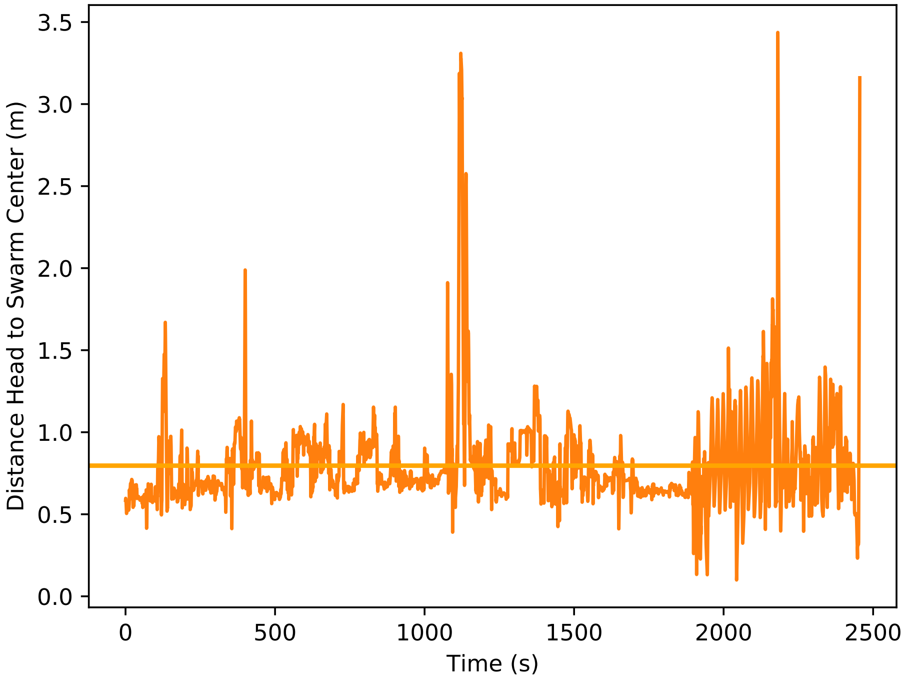


Figure A.44: Distance (in meters) from the head (HoloLens) to the agent's swarm center during the performance session for *User 3*.

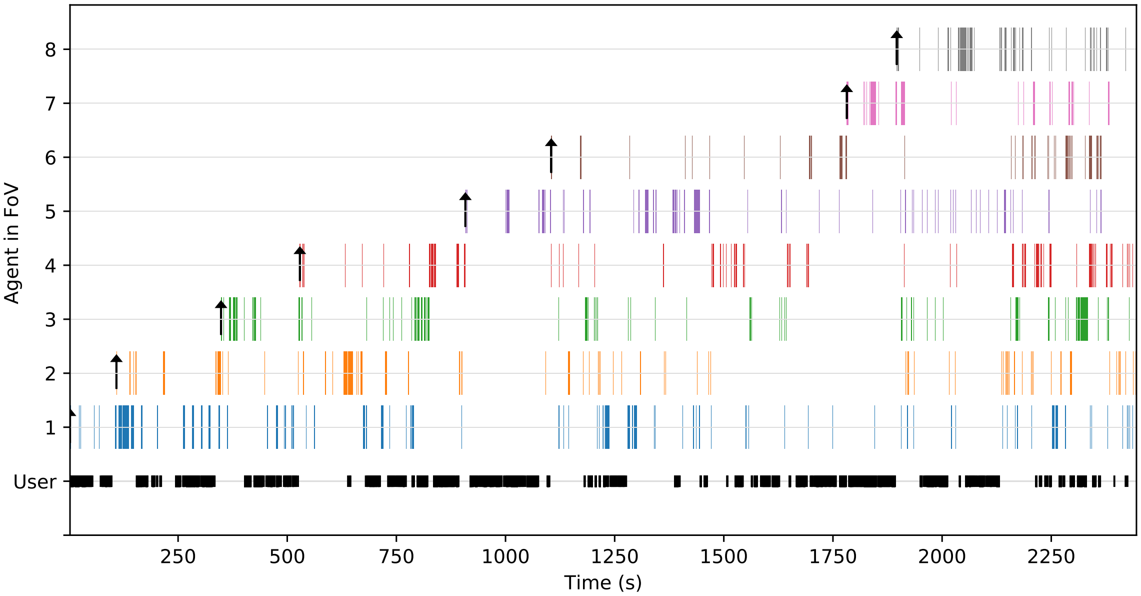


Figure A.45: Periods when agents were in the HoloLen's *field of view* (fov) during the performance session for *User 3*. The first row 'User' represents the user activity in the MIDI controller along the session. The arrows represent the moment of the first appearance of an agent.

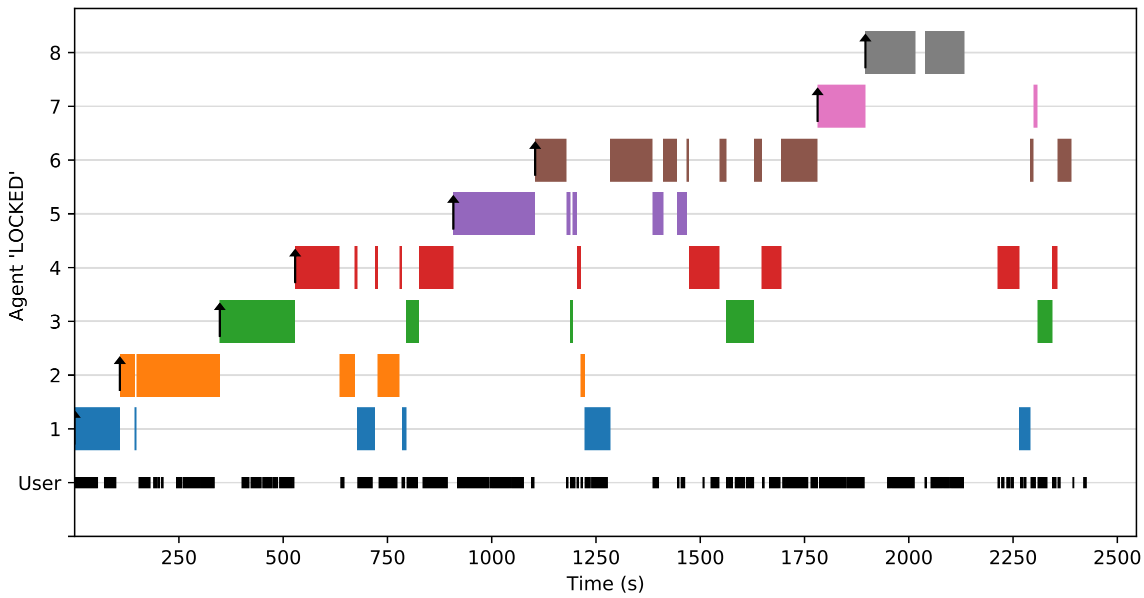


Figure A.46: Periods when agents were ’LOCKED’ during the performance session for *User 3*. The first row ’User’ represents the user activity in the MIDI controller along the session. The arrows represent the moment of the first appearance of an agent.

A.4.3 Agents Movement in a Session

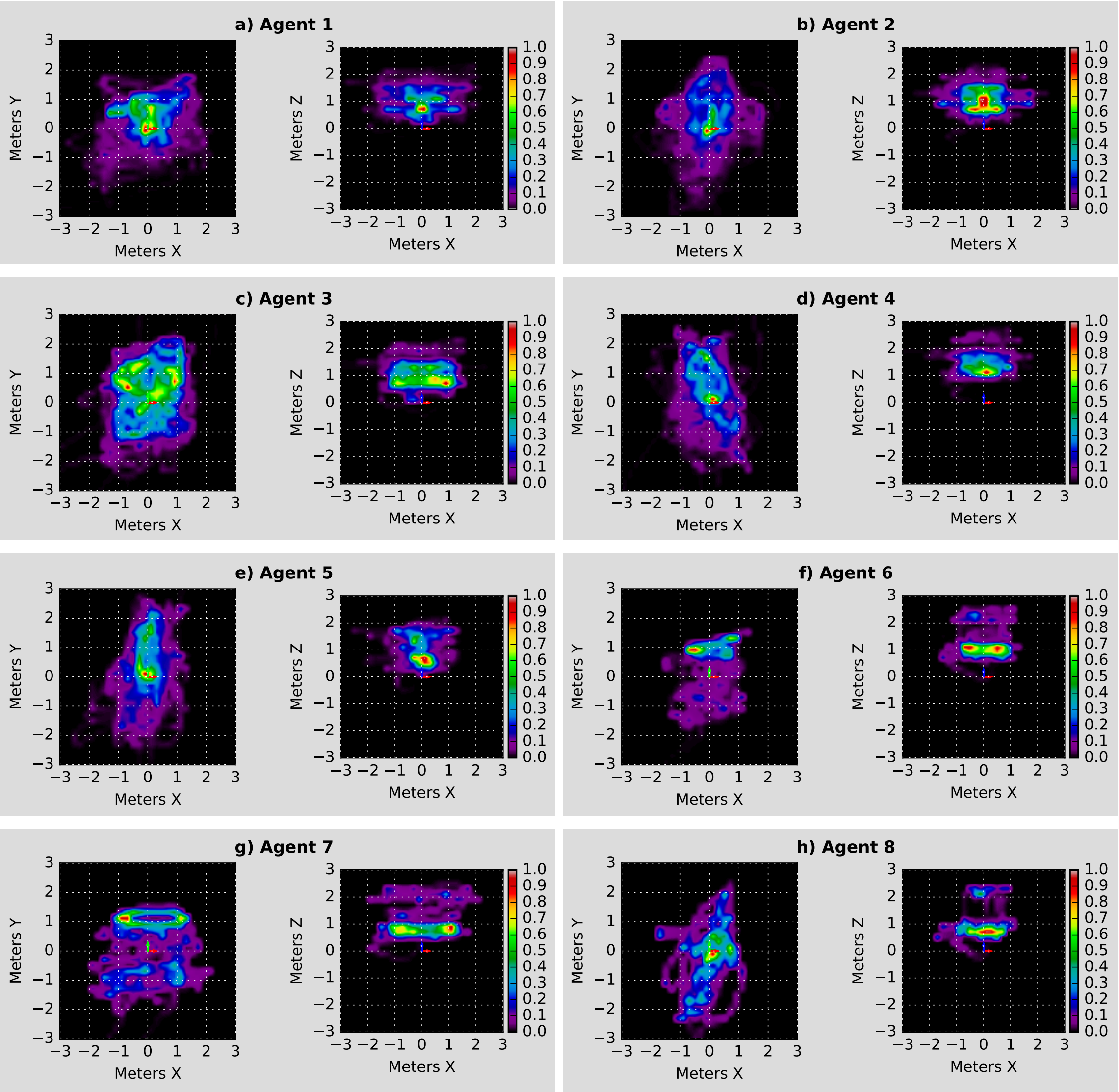


Figure A.47: Heatmaps for the movement of every agent during the performance session for *User 3*. It shows the locations where they spent most of their time, as well as trajectories. The view is from top (x, y) and from back (x, z) per each agent.

A.4.4 Tempo Changes

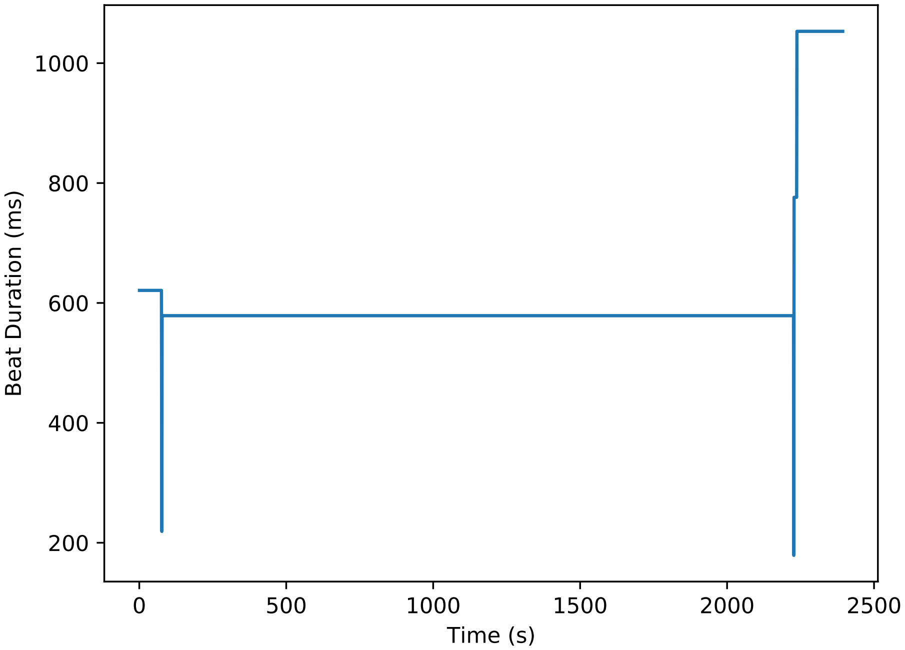


Figure A.48: Beat duration period for the metronome during the performance session for *User3*. This is limited to periods of 5000 ms to detect the actual BPM configuration given by the user at any point in time. The high impulses correspond to periods where the user changed the tempo or just stopped the metronome, wait for a moment, and played it again.

A.5 User 4

A.5.1 The User and the Physical Space

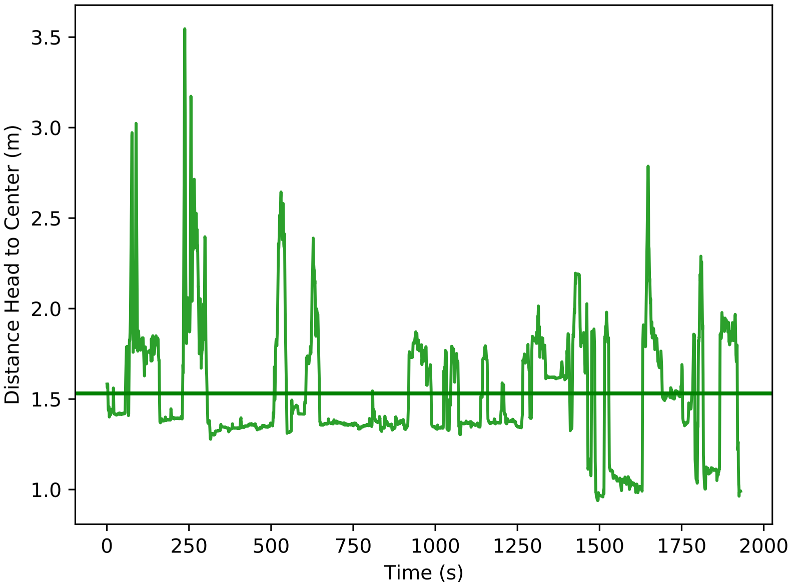


Figure A.49: Distance (in meters) from the head (HoloLens) to the center of the room during the performance session for *User 4*

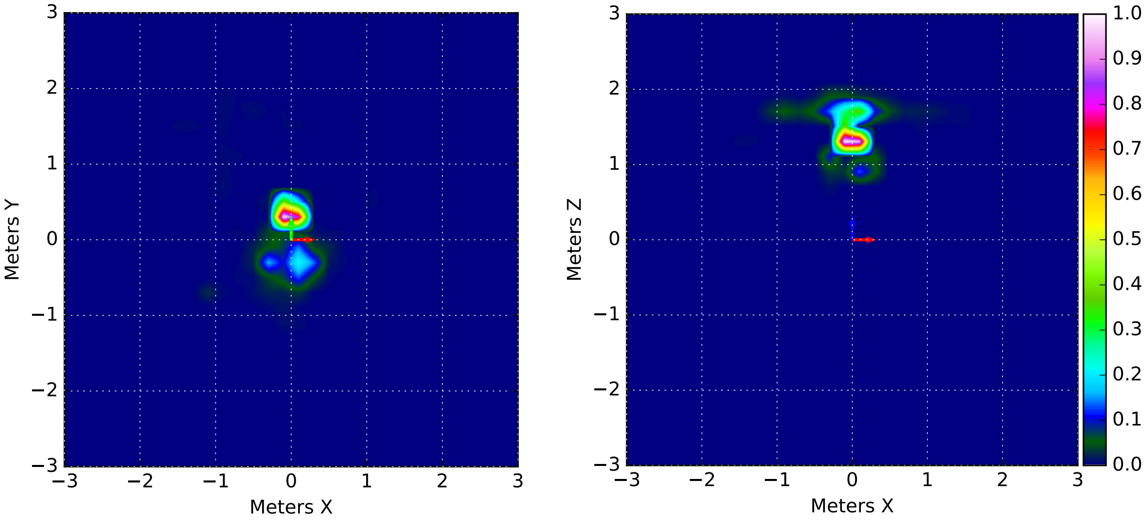


Figure A.50: Locations where *User 4* spent most of his or her time during the performance session. This view is from top (x, y) and from back (x, z).

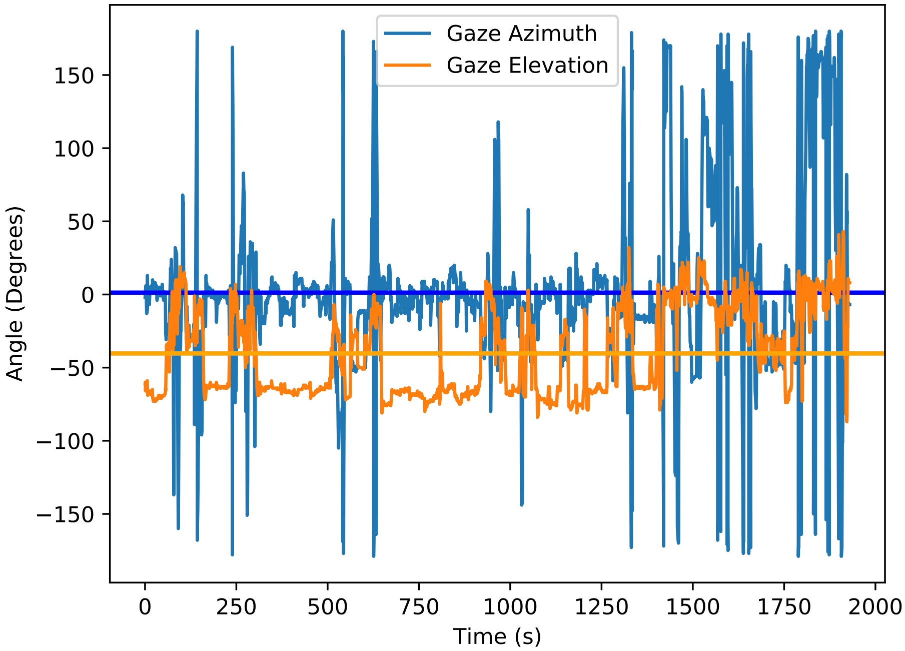


Figure A.51: Azimuth and Elevation angles (in degrees) for gaze direction regarding *User 4* during the performance session.

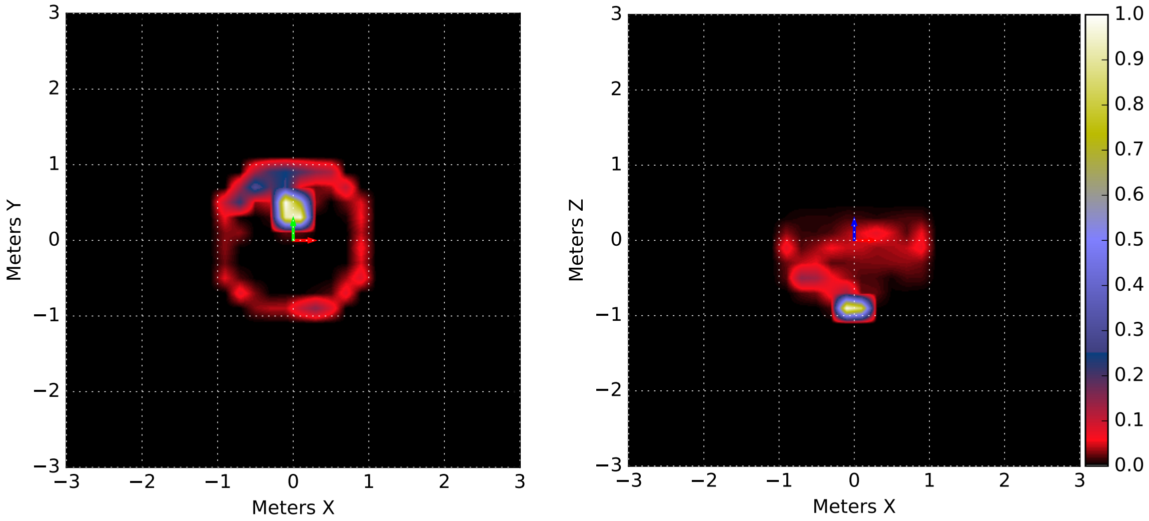


Figure A.52: Directions where *User 4* spent most of his or her time during the performance session. This view is from top (x, y) and from back (x, z). As this is a unit vector, the planes are just a reference and does not represent the actual room and its center.

A.5.2 The User and the Rigid Body

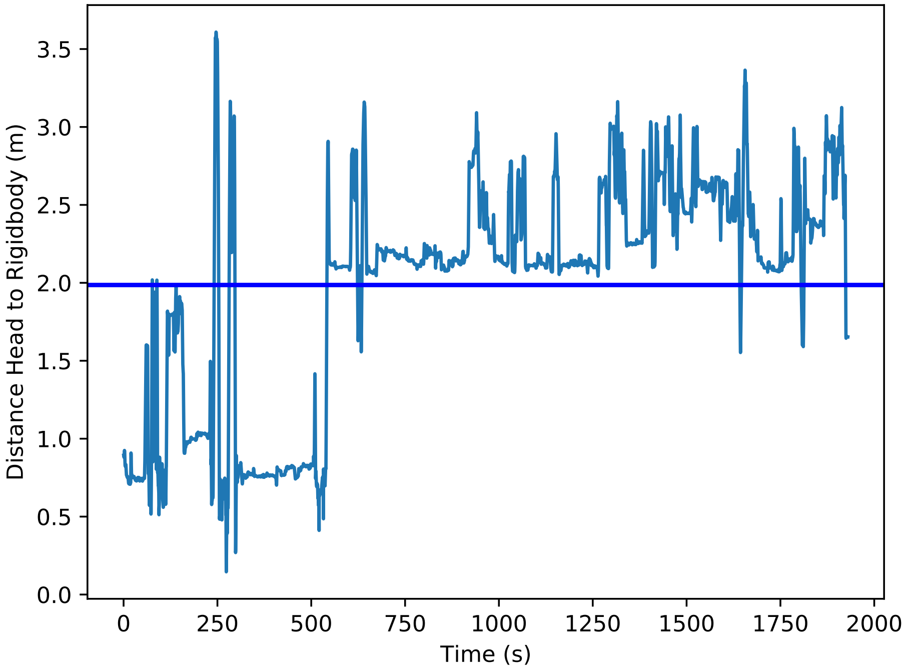


Figure A.53: Distance (in meters) from the head (HoloLens) to the rigid body during the performance session for *User 4*.

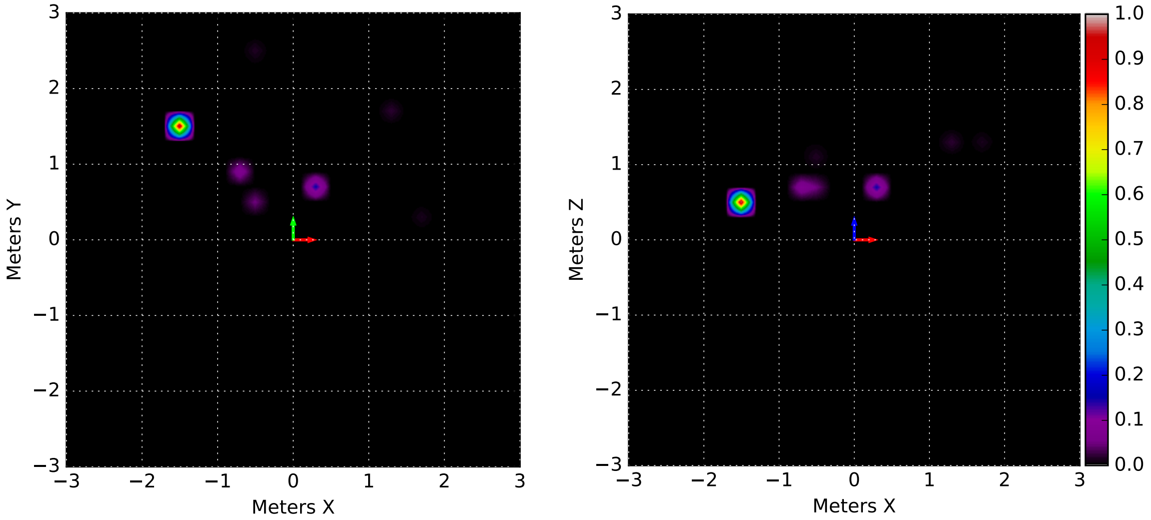


Figure A.54: Frequent locations where *User 4* placed the rigid body during the performance session. This view is from top (x, y) and from back (x, z).

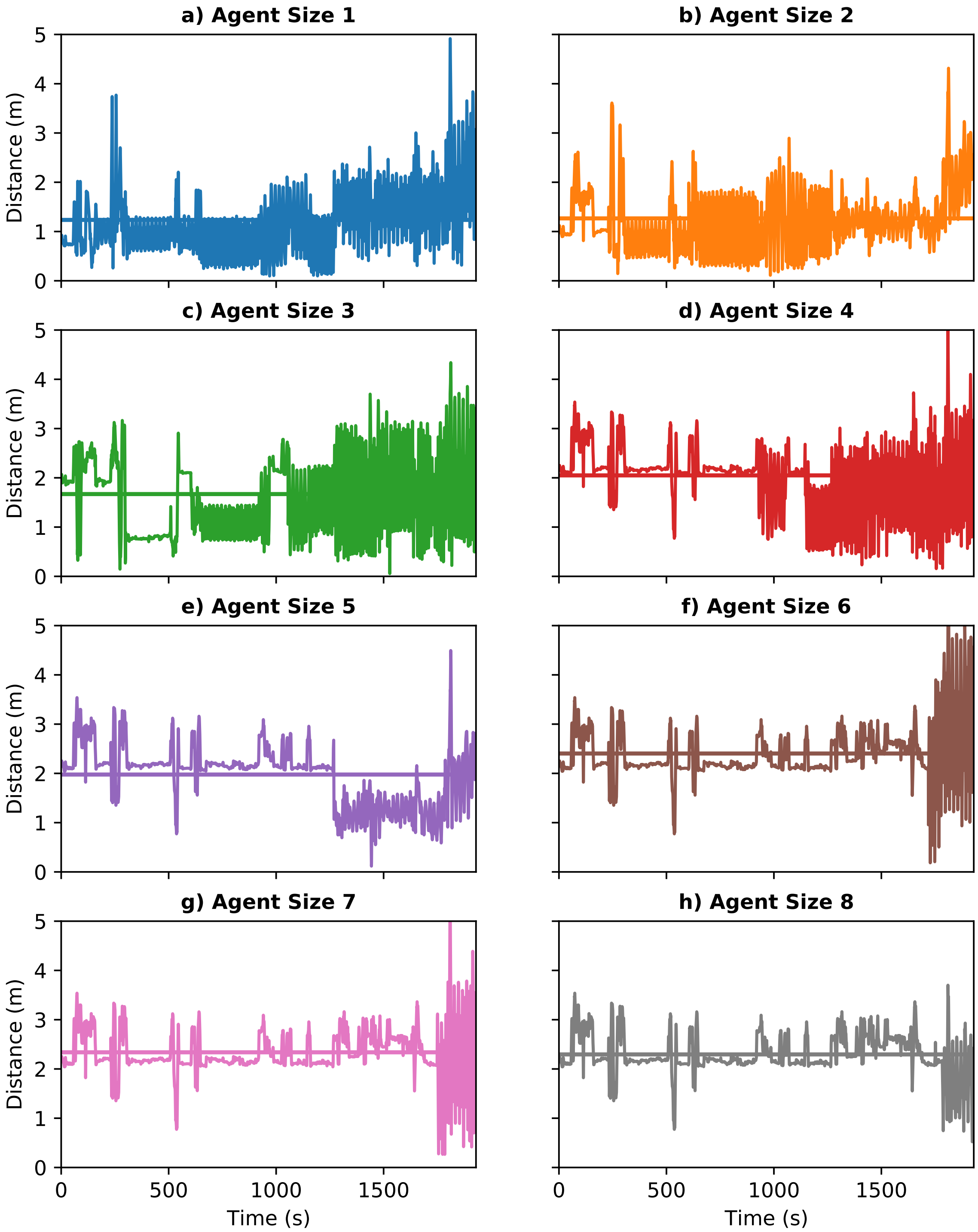


Figure A.55: Distance (in meters) from the head (HoloLens) to every agent during the performance session for *User 4*.

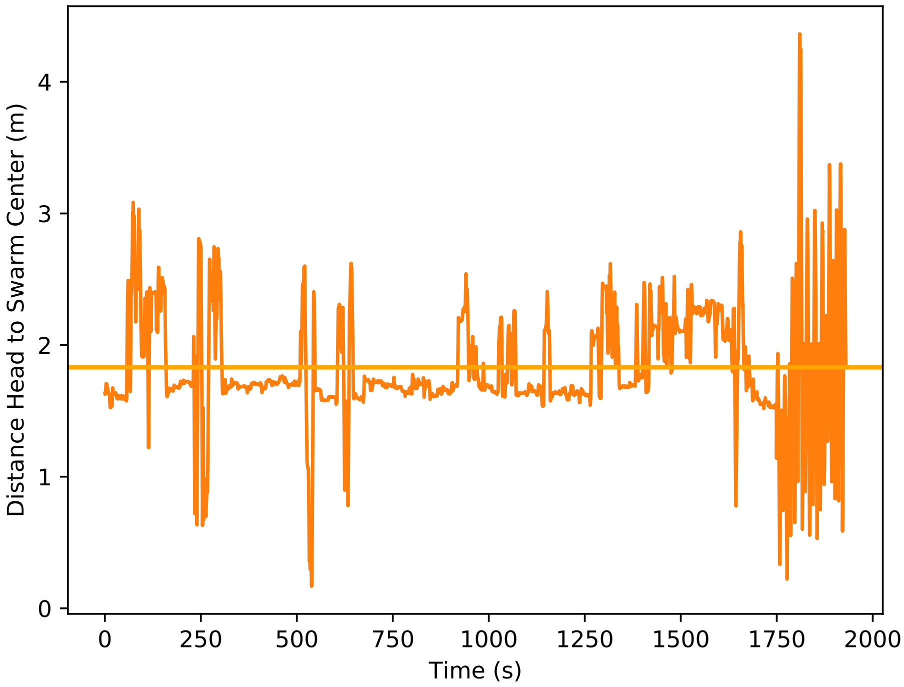


Figure A.56: Distance (in meters) from the head (HoloLens) to the agent's swarm center during the performance session for *User 4*.

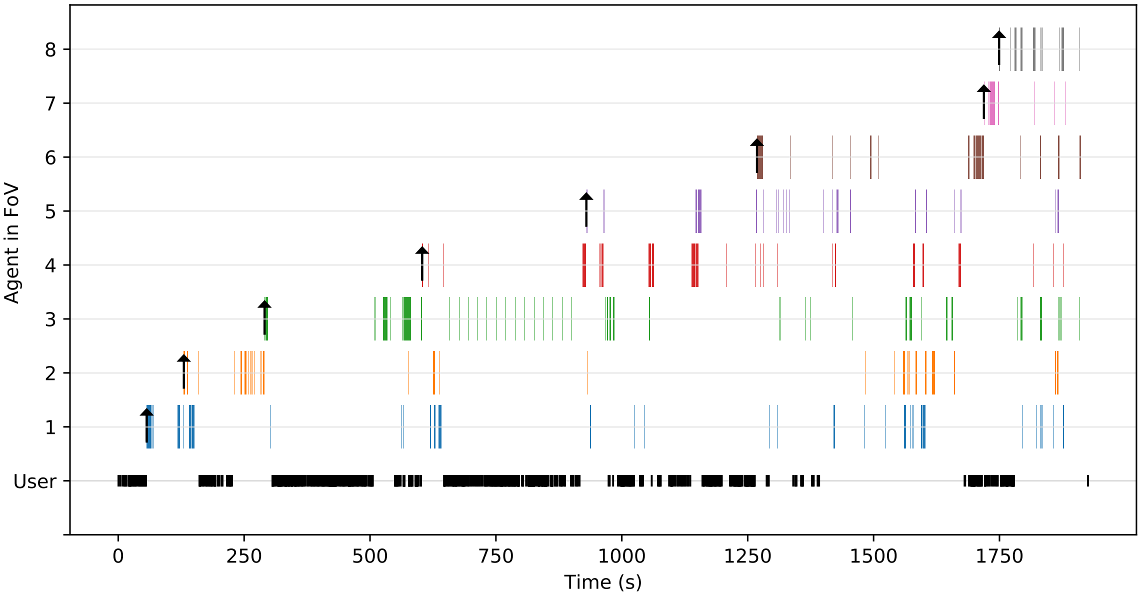


Figure A.57: Periods when agents were in the HoloLen's *field of view* (fov) during the performance session for *User 4*. The first row 'User' represents the user activity in the MIDI controller along the session. The arrows represent the moment of the first appearance of an agent.

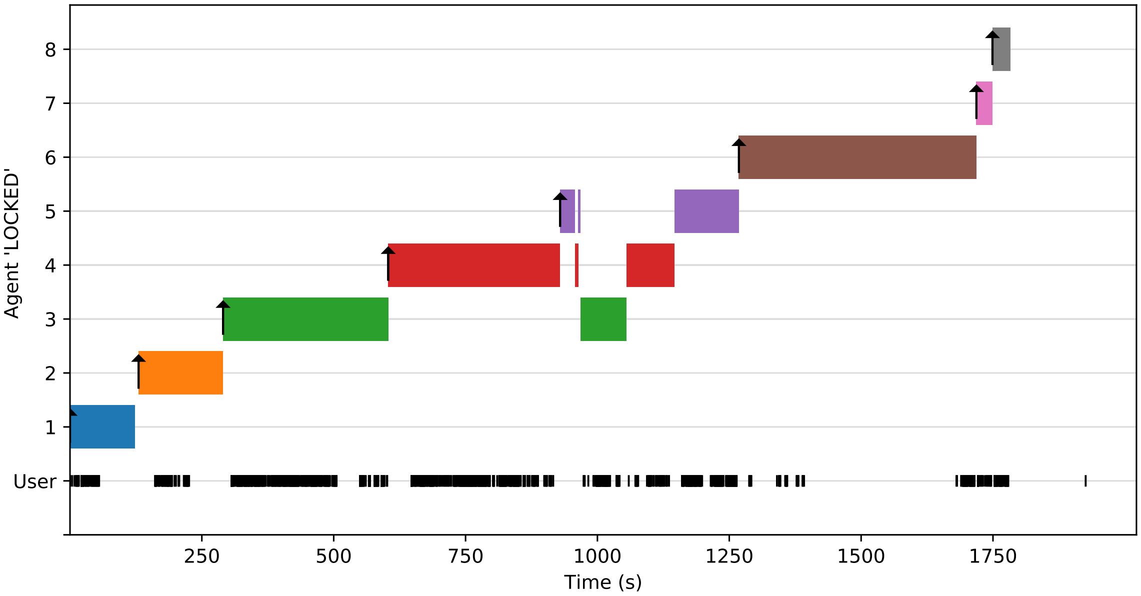


Figure A.58: Periods when agents were 'LOCKED' during the performance session for *User 4*. The first row 'User' represents the user activity in the MIDI controller along the session. The arrows represent the moment of the first appearance of an agent.

A.5.3 Agents Movement in a Session

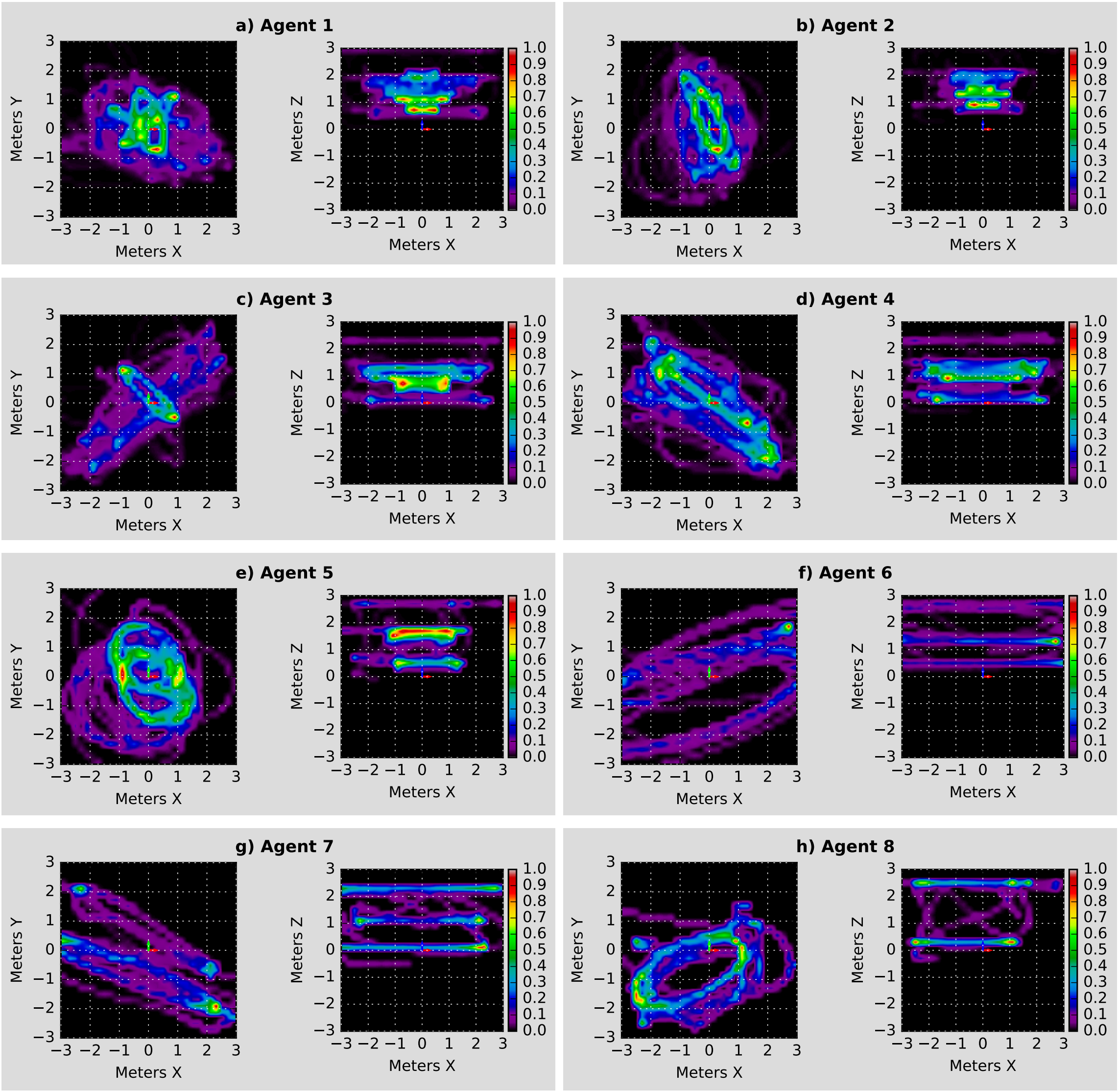


Figure A.59: Heatmaps for the movement of every agent during the performance session for *User 4*. It shows the locations where they spent most of their time, as well as trajectories. The view is from top (x, y) and from back (x, z) per each agent.

A.5.4 Tempo Changes

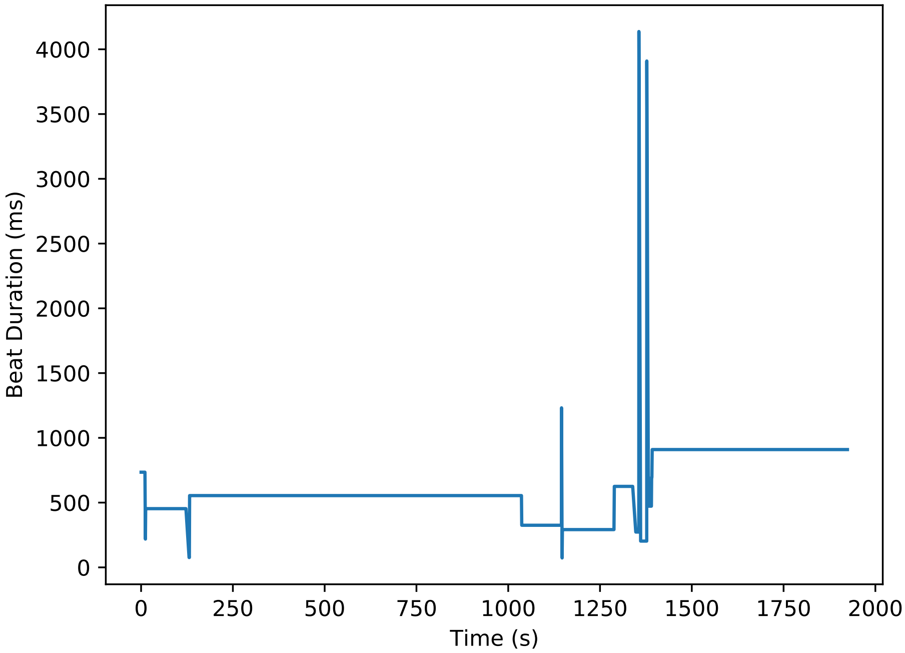


Figure A.60: Beat duration period for the metronome during the performance session for *User4*. This is limited to periods of 5000 ms to detect the actual BPM configuration given by the user at any point in time. The high impulses correspond to periods where the user changed the tempo or just stopped the metronome, wait for a moment, and played it again.

A.6 User 5

A.6.1 The User and the Physical Space

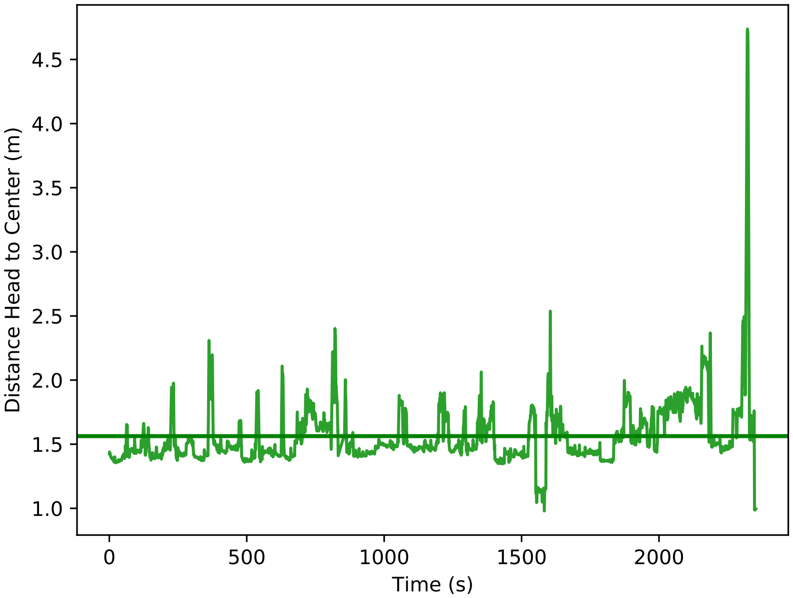


Figure A.61: Distance (in meters) from the head (HoloLens) to the center of the room during the performance session for *User 5*

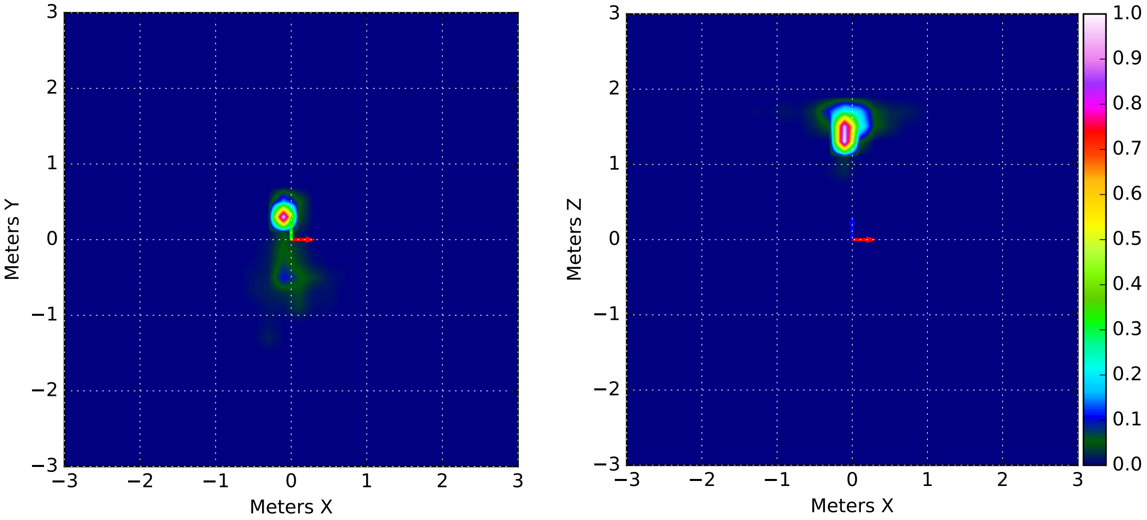


Figure A.62: Locations where *User 5* spent most of his or her time during the performance session. This view is from top (x, y) and from back (x, z).

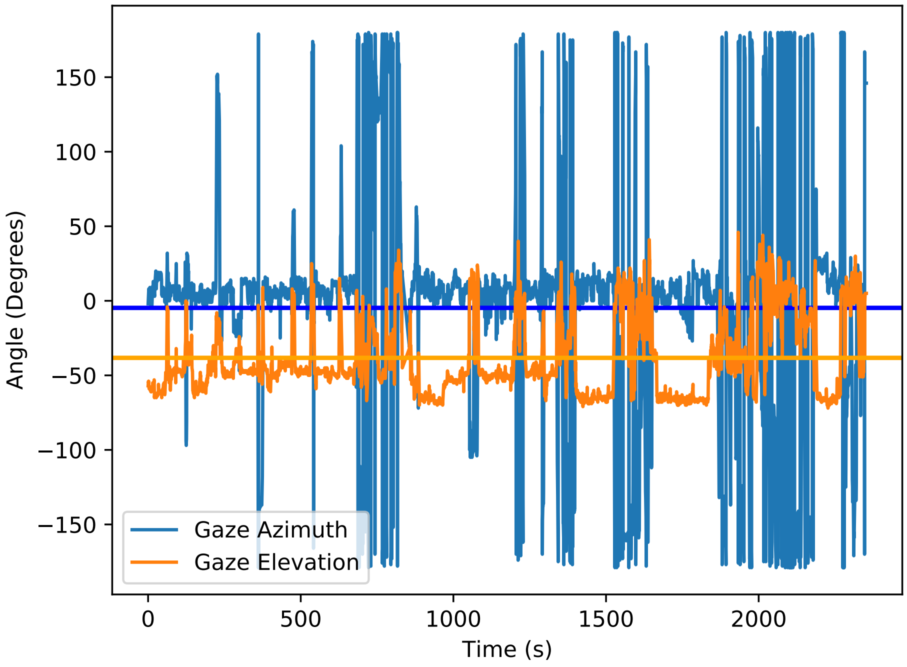


Figure A.63: Azimuth and Elevation angles (in degrees) for gaze direction regarding *User 5* during the performance session.

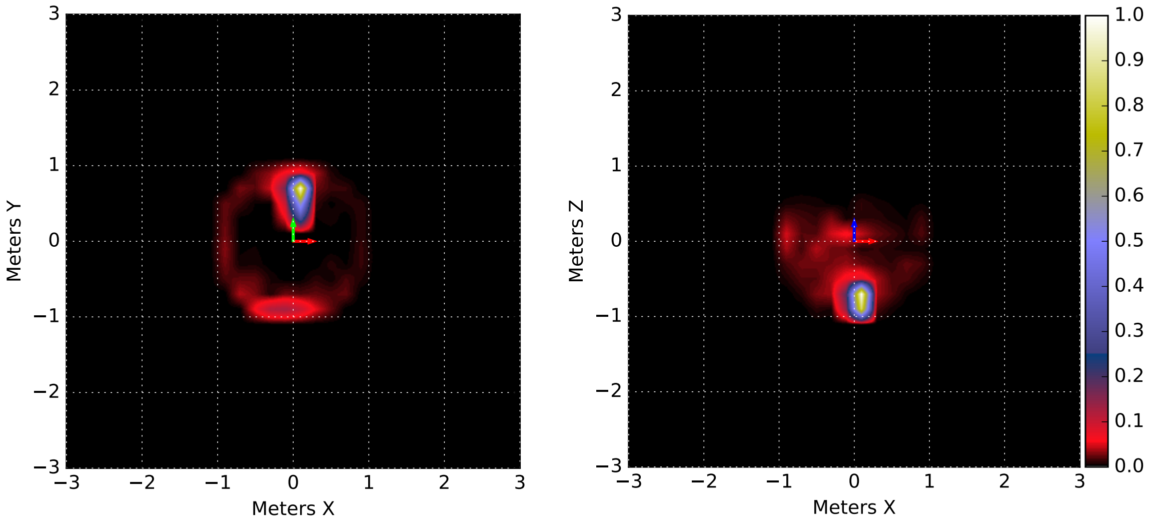


Figure A.64: Directions where *User 5* spent most of his or her time during the performance session. This view is from top (x, y) and from back (x, z). As this is a unit vector, the planes are just a reference and does not represent the actual room and its center.

A.6.2 The User and the Rigid Body

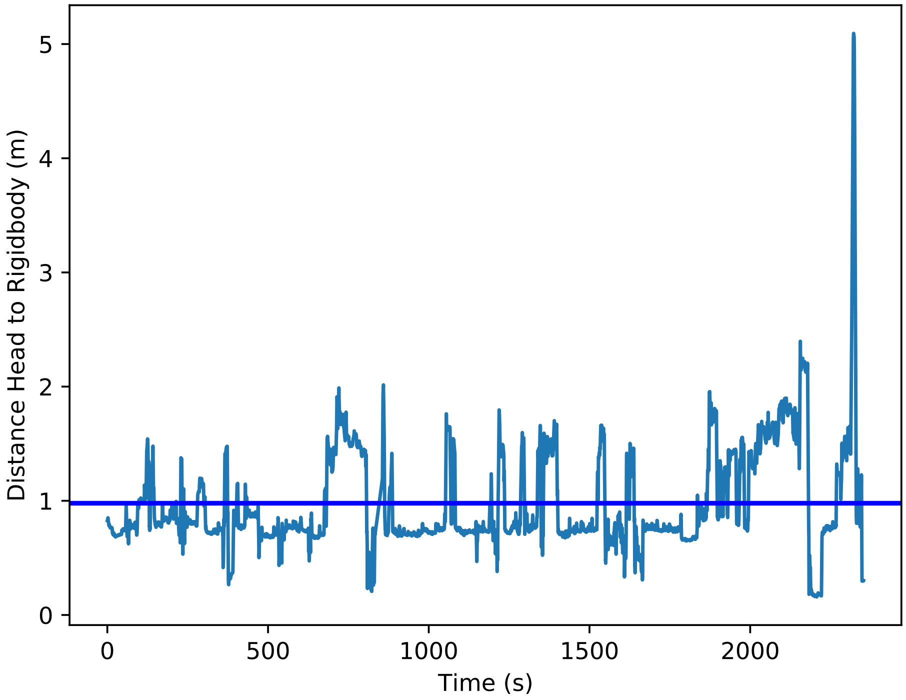


Figure A.65: Distance (in meters) from the head (HoloLens) to the rigid body during the performance session for *User 5*.

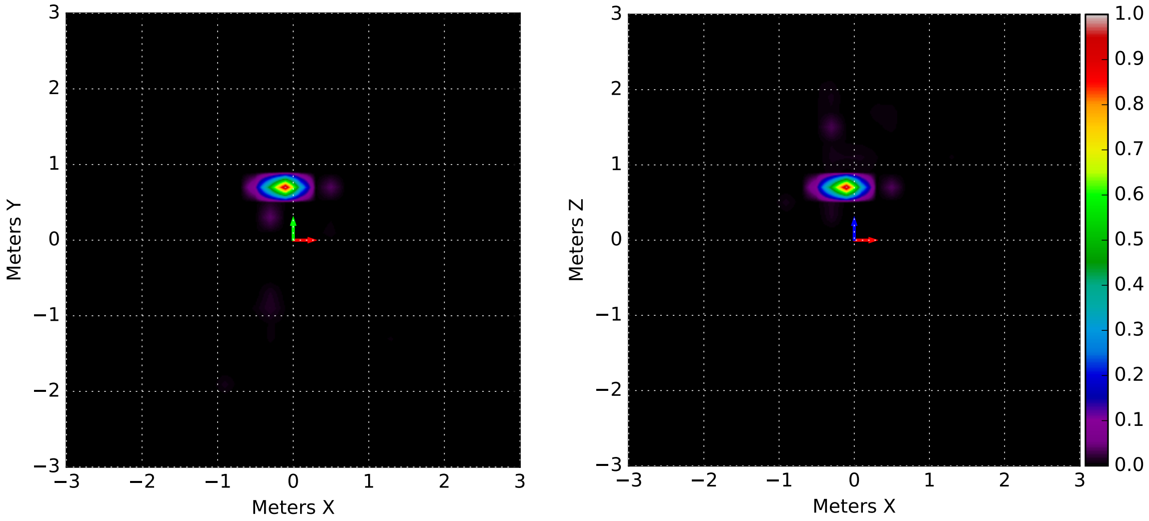


Figure A.66: Frequent locations where *User 5* placed the rigid body during the performance session. This view is from top (x, y) and from back (x, z).

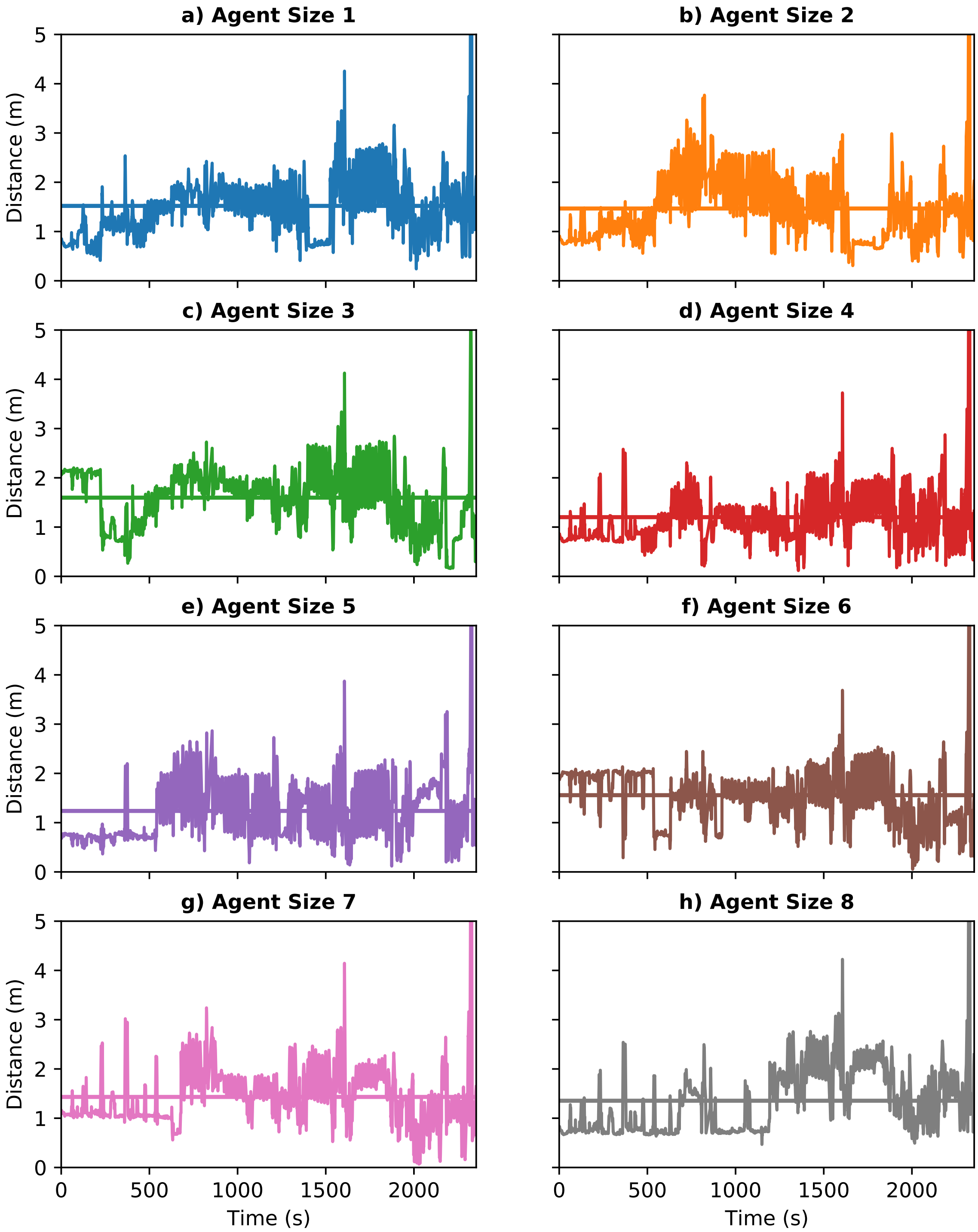


Figure A.67: Distance (in meters) from the head (HoloLens) to every agent during the performance session for *User 5*.

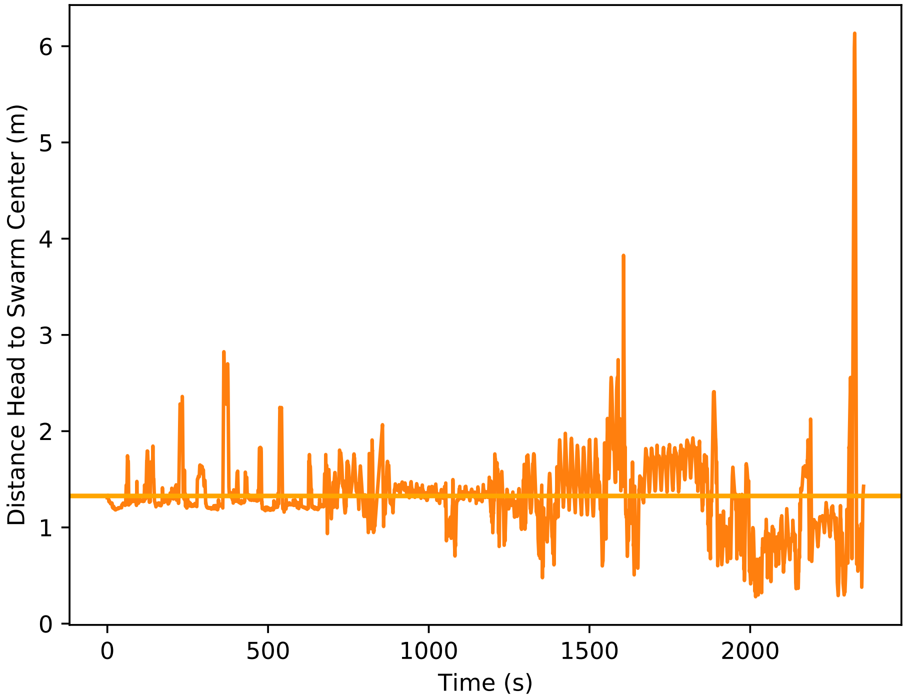


Figure A.68: Distance (in meters) from the head (HoloLens) to the agent's swarm center during the performance session for *User 5*.

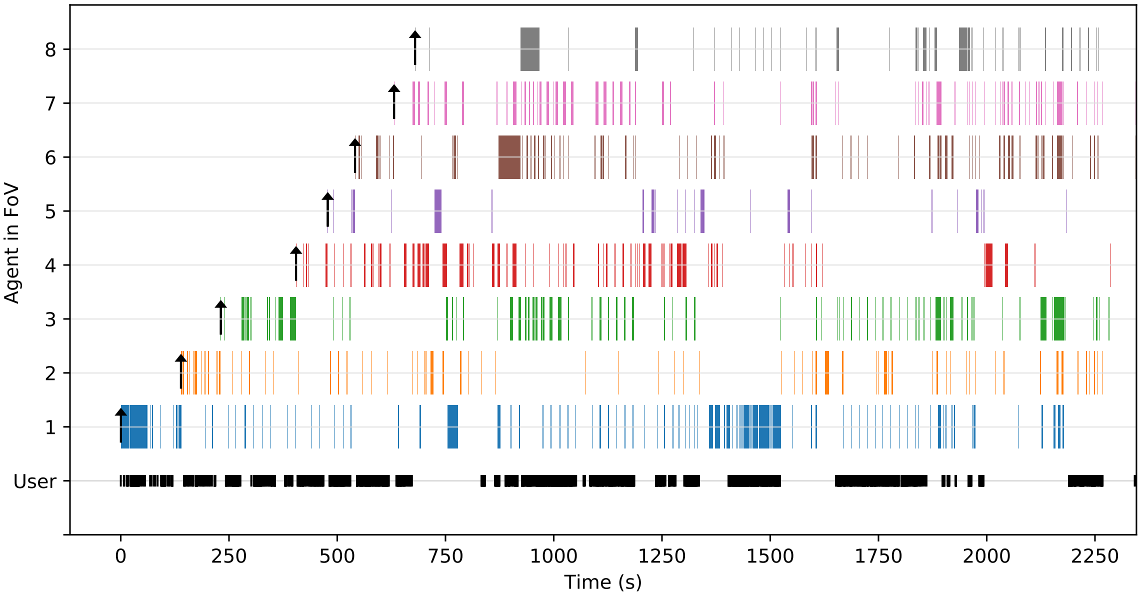


Figure A.69: Periods when agents were in the HoloLen's *field of view* (fov) during the performance session for *User 5*. The first row 'User' represents the user activity in the MIDI controller along the session. The arrows represent the moment of the first appearance of an agent.

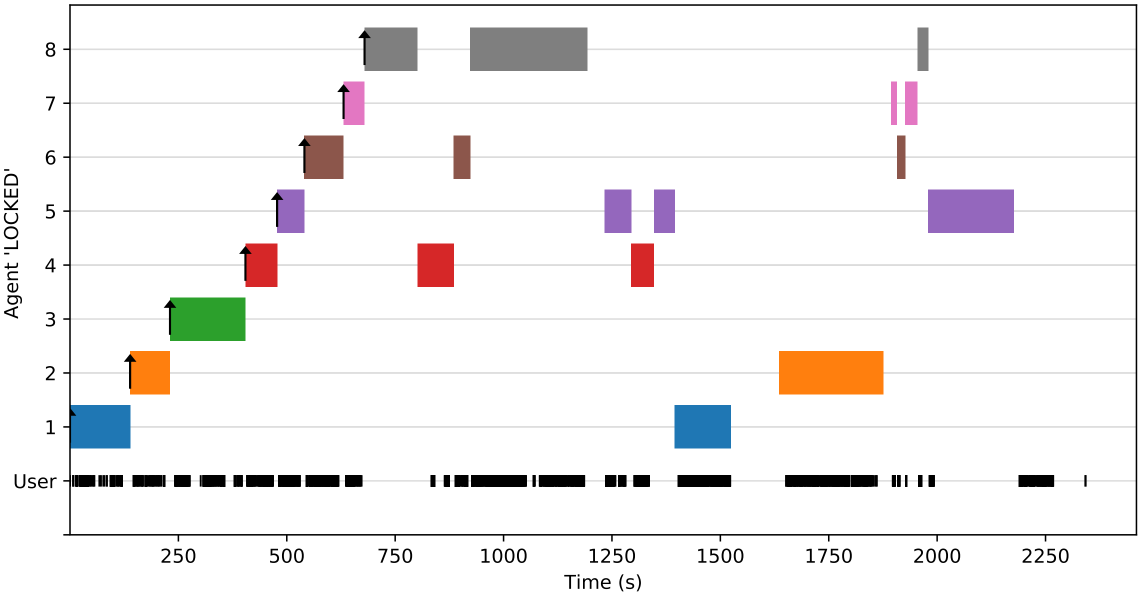


Figure A.70: Periods when agents were 'LOCKED' during the performance session for *User 5*. The first row 'User' represents the user activity in the MIDI controller along the session. The arrows represent the moment of the first appearance of an agent.

A.6.3 Agents Movement in a Session

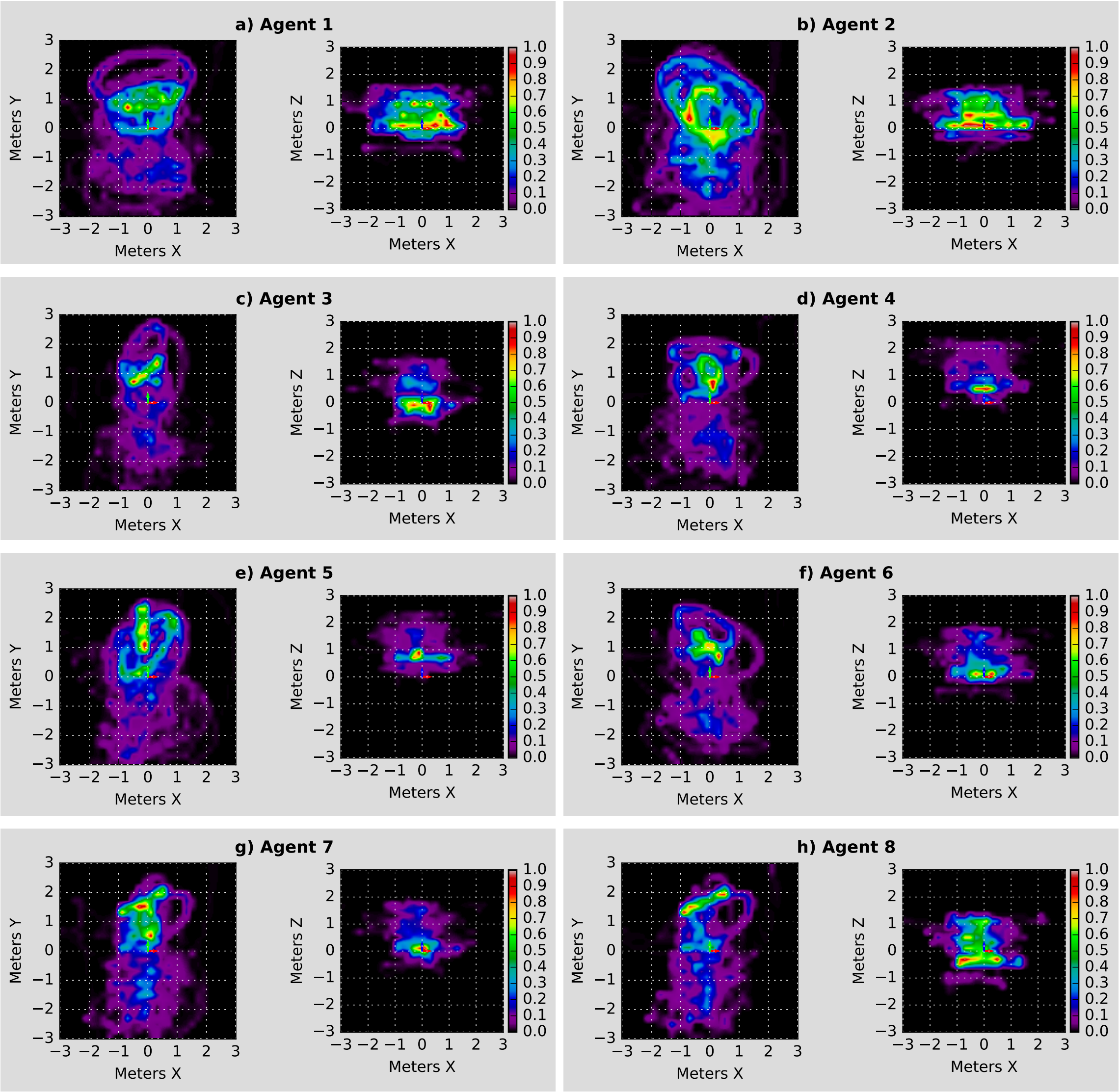


Figure A.71: Heatmaps for the movement of every agent during the performance session for *User 5*. It shows the locations where they spent most of their time, as well as trajectories. The view is from top (x, y) and from back (x, z) per each agent.

A.6.4 Tempo Changes

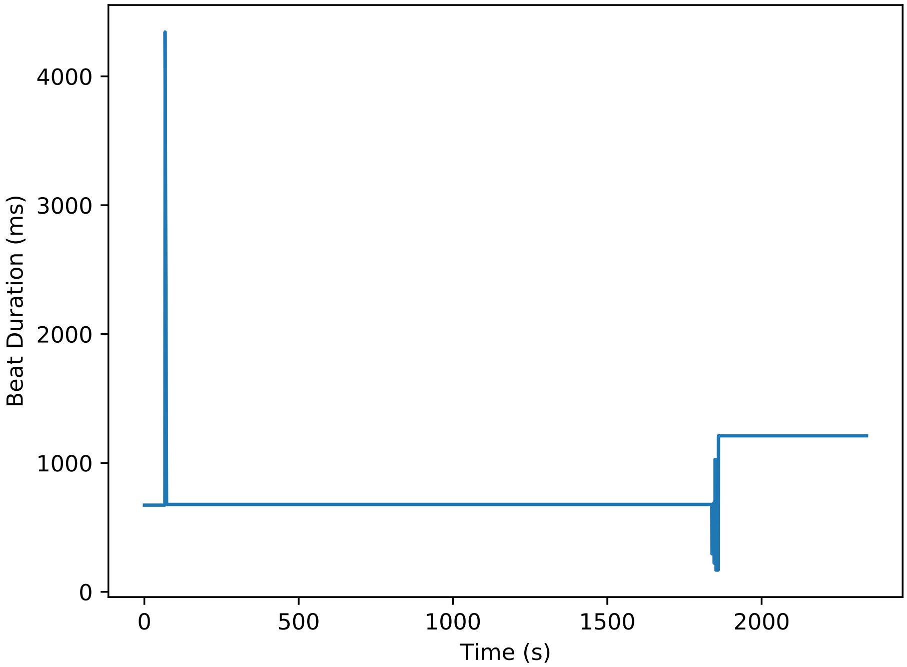


Figure A.72: Beat duration period for the metronome during the performance session for *User5*. This is limited to periods of 5000 ms to detect the actual BPM configuration given by the user at any point in time. The high impulses correspond to periods where the user changed the tempo or just stopped the metronome, wait for a moment, and played it again.

A.7 User 6

A.7.1 The User and the Physical Space

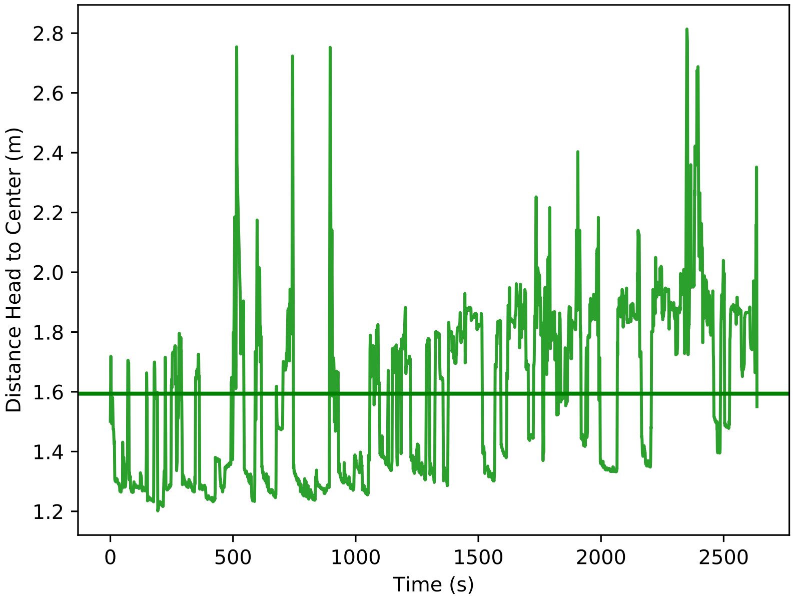


Figure A.73: Distance (in meters) from the head (HoloLens) to the center of the room during the performance session for *User 6*

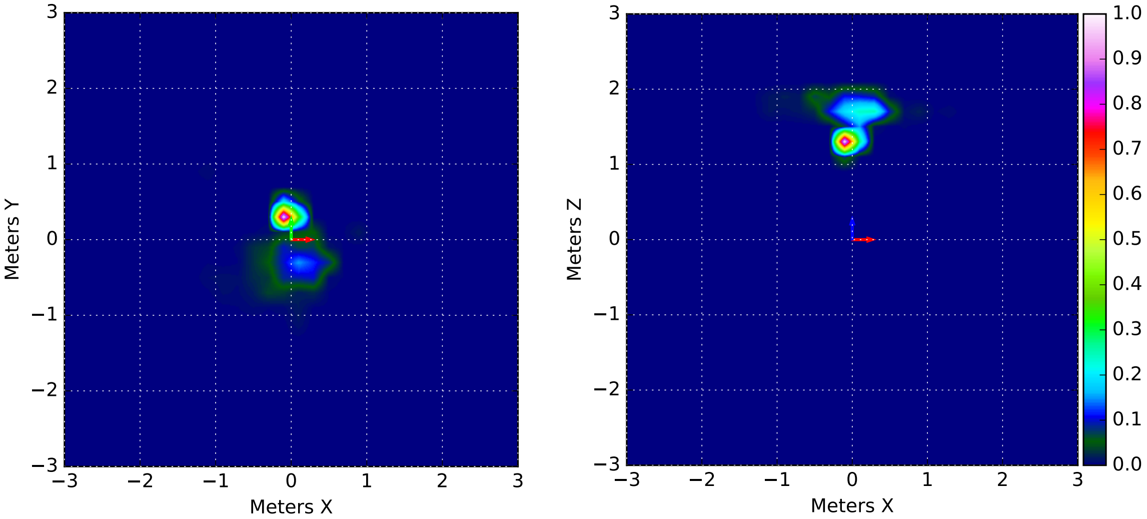


Figure A.74: Locations where *User 6* spent most of his or her time during the performance session. This view is from top (x, y) and from back (x, z).

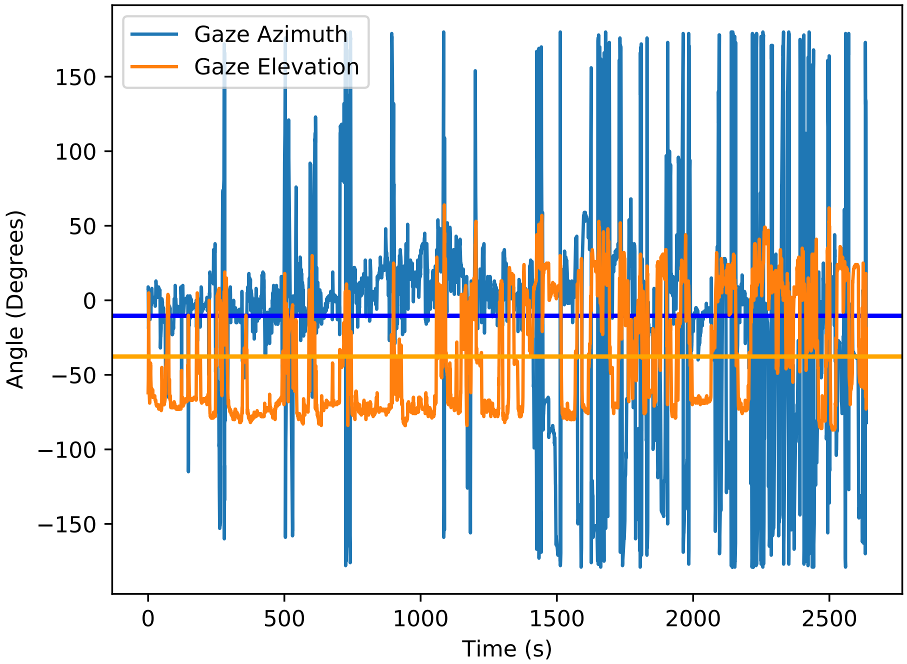


Figure A.75: Azimuth and Elevation angles (in degrees) for gaze direction regarding *User 6* during the performance session.

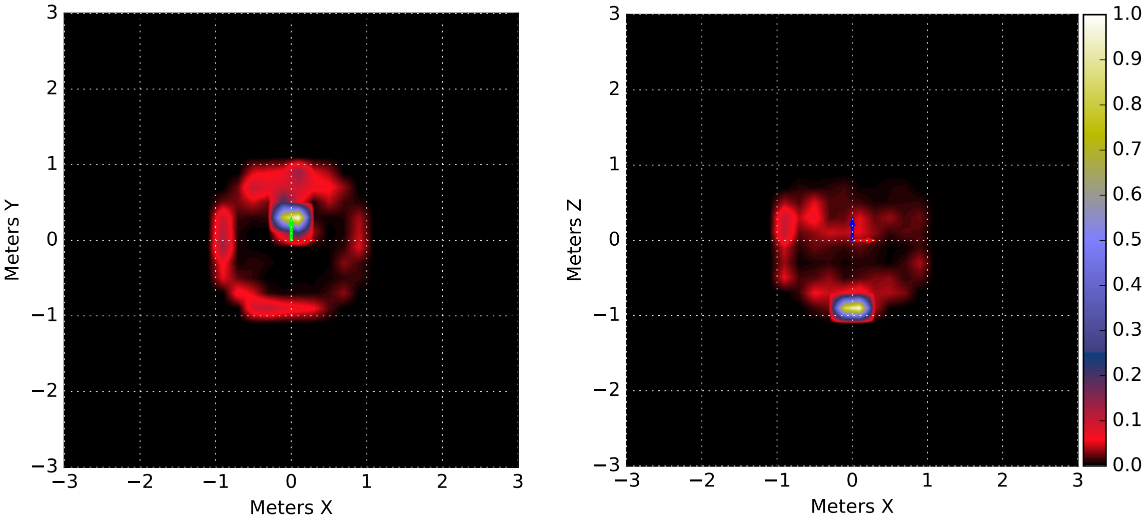


Figure A.76: Directions where *User 6* spent most of his or her time during the performance session. This view is from top (x, y) and from back (x, z). As this is a unit vector, the planes are just a reference and does not represent the actual room and its center.

A.7.2 The User and the Rigid Body

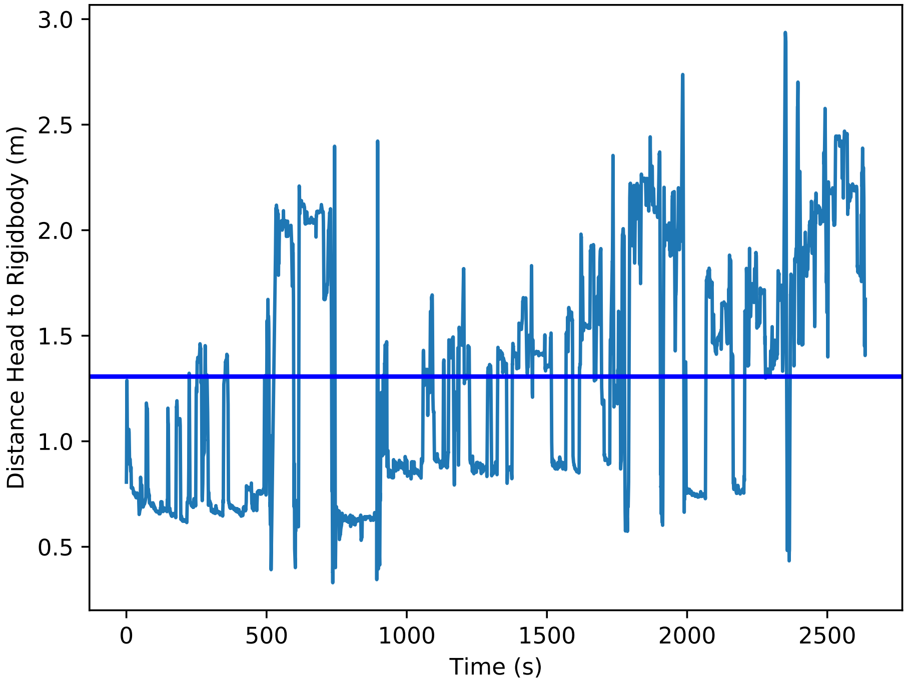


Figure A.77: Distance (in meters) from the head (HoloLens) to the rigid body during the performance session for *User 6*.

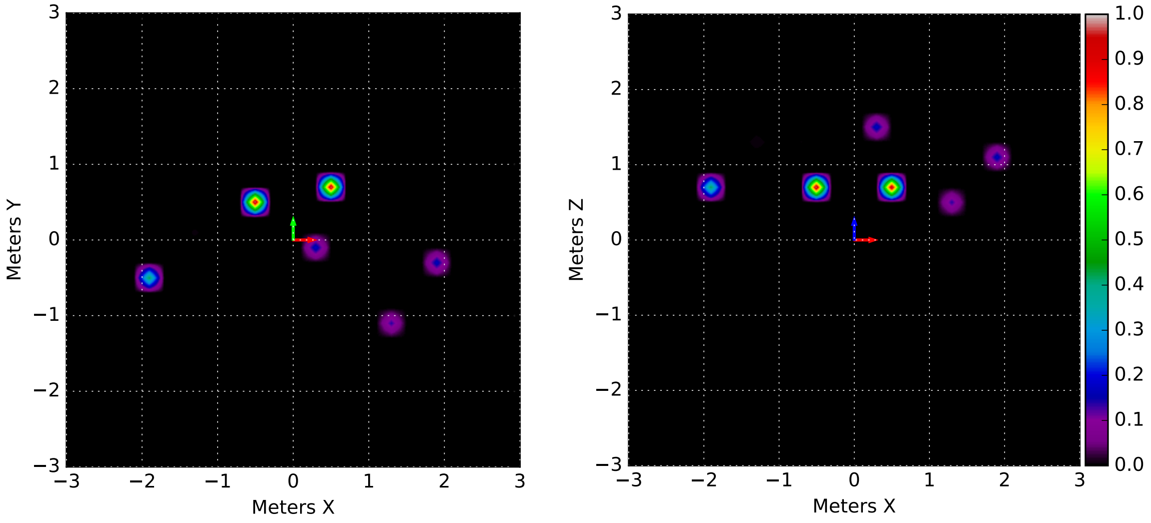


Figure A.78: Frequent locations where *User 6* placed the rigid body during the performance session. This view is from top (x, y) and from back (x, z).

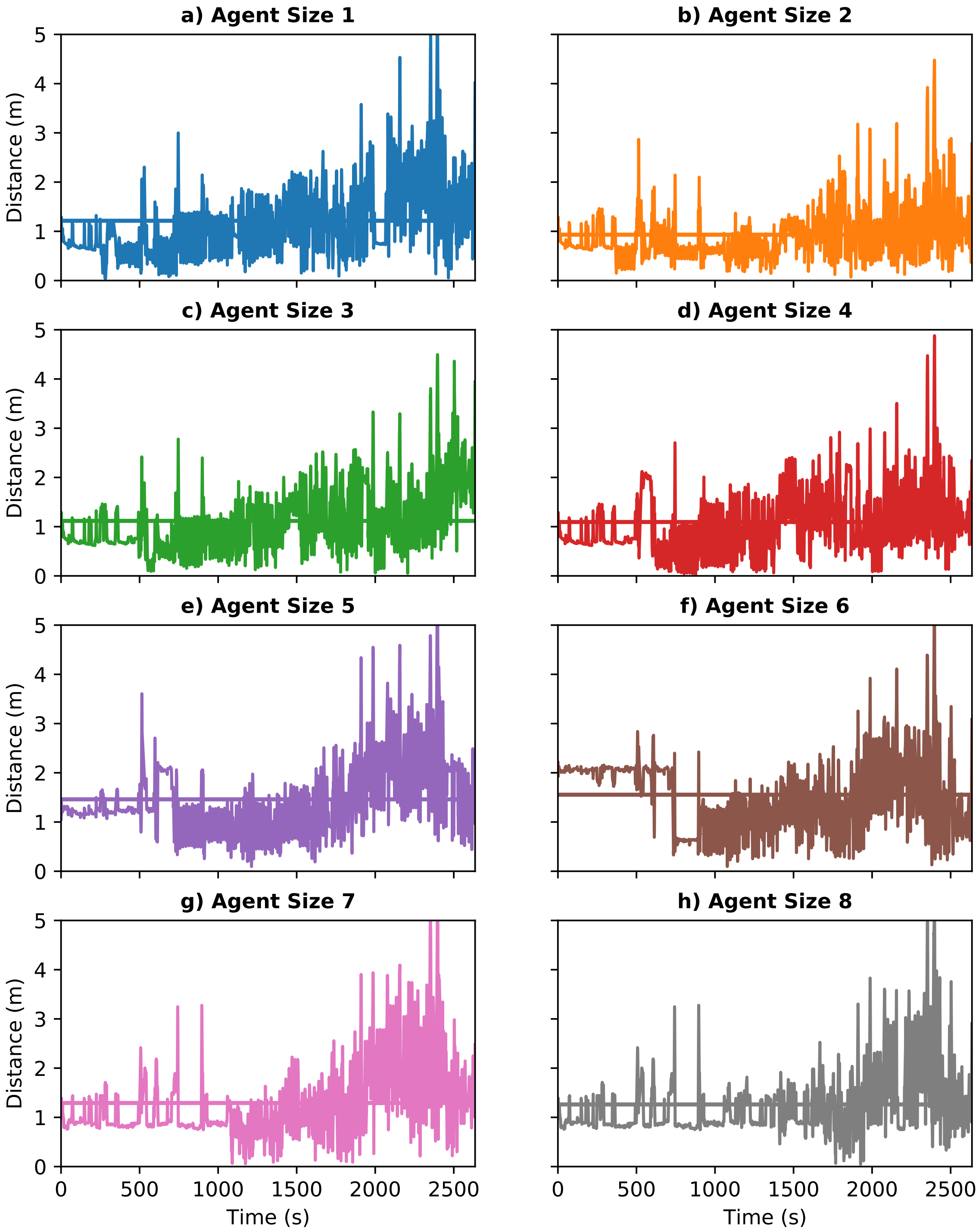


Figure A.79: Distance (in meters) from the head (HoloLens) to every agent during the performance session for *User 6*.

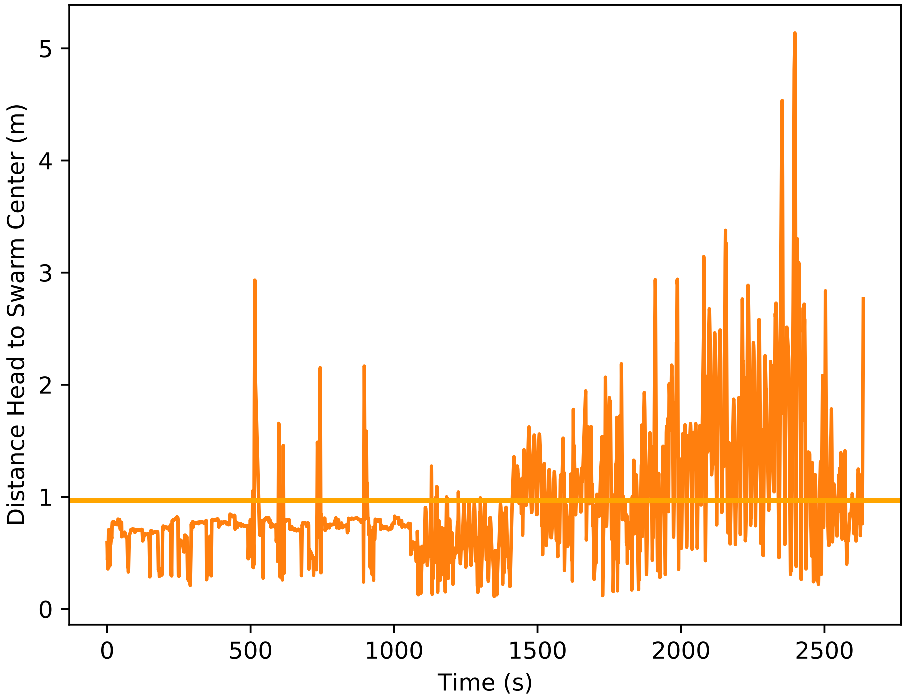


Figure A.80: Distance (in meters) from the head (HoloLens) to the agent's swarm center during the performance session for *User 6*.

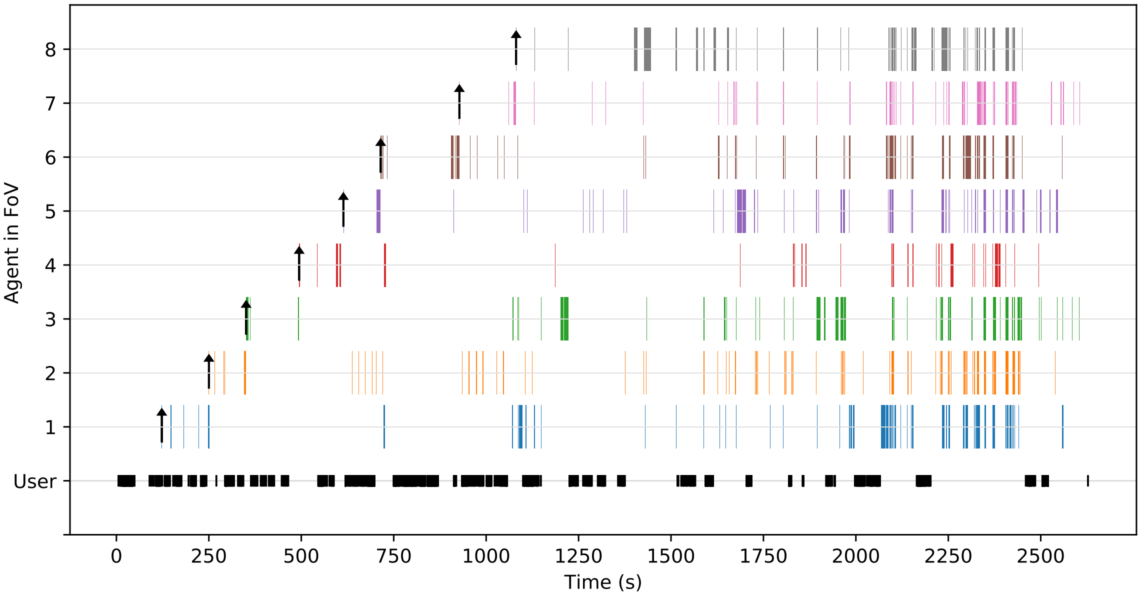


Figure A.81: Periods when agents were in the HoloLen's *field of view* (fov) during the performance session for *User 6*. The first row 'User' represents the user activity in the MIDI controller along the session. The arrows represent the moment of the first appearance of an agent.

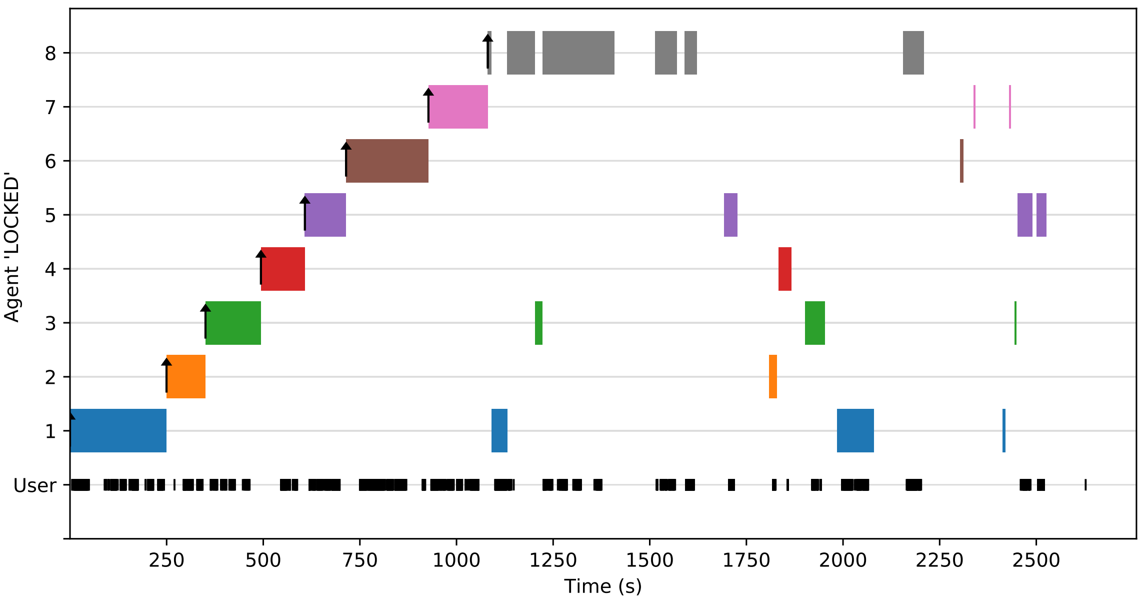


Figure A.82: Periods when agents were 'LOCKED' during the performance session for *User 6*. The first row 'User' represents the user activity in the MIDI controller along the session. The arrows represent the moment of the first appearance of an agent.

A.7.3 Agents Movement in a Session

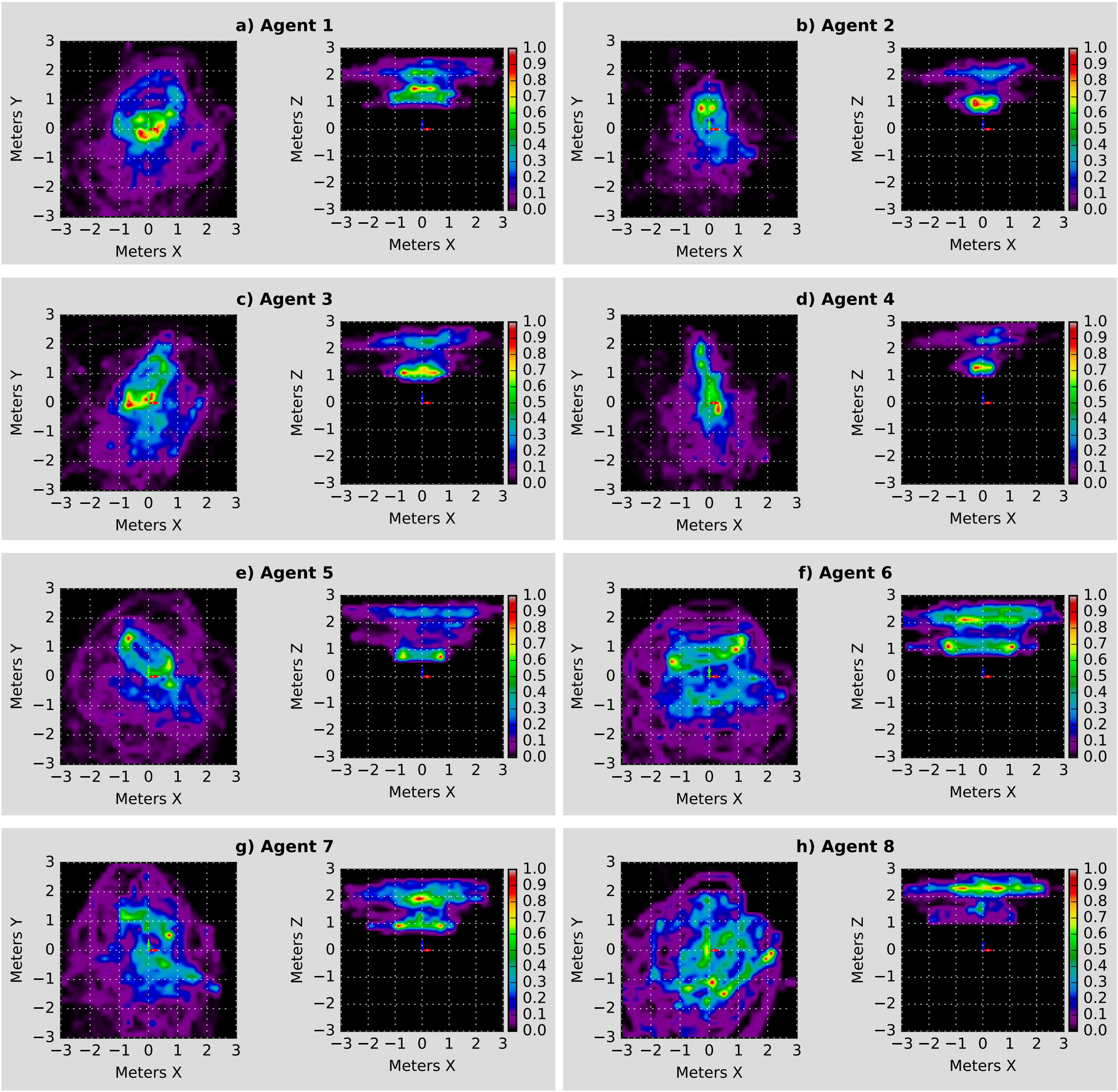


Figure A.83: Heatmaps for the movement of every agent during the performance session for *User 6*. It shows the locations where they spent most of their time, as well as trajectories. The view is from top (x, y) and from back (x, z) per each agent.

A.7.4 Tempo Changes

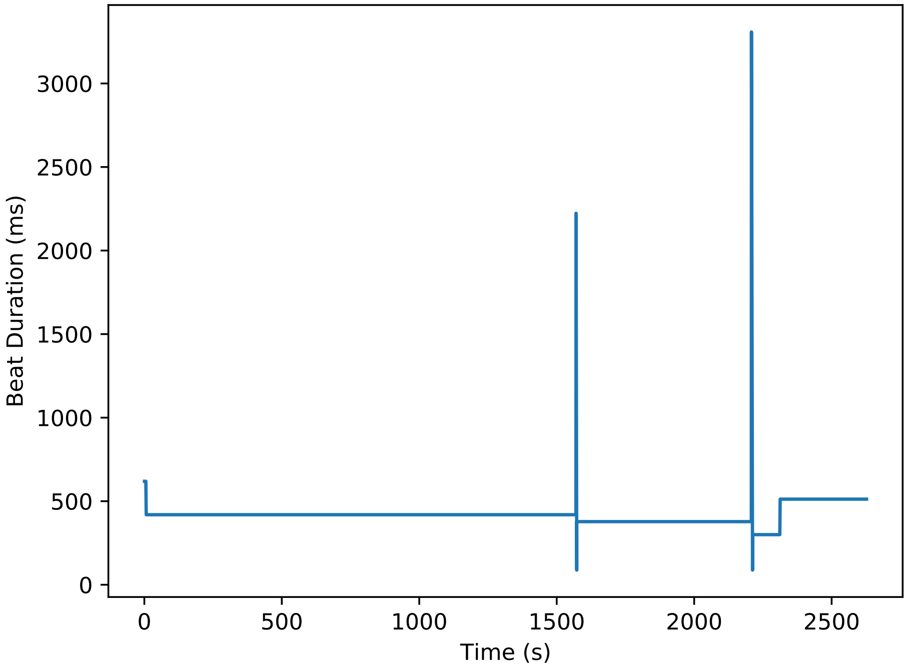


Figure A.84: Beat duration period for the metronome during the performance session for *User6*. This is limited to periods of 5000 ms to detect the actual BPM configuration given by the user at any point in time. The high impulses correspond to periods where the user changed the tempo or just stopped the metronome, wait for a moment, and played it again.

Appendix B

User Survey

B.1 Musical Background

B.1.1 Do you have a formal education in music?

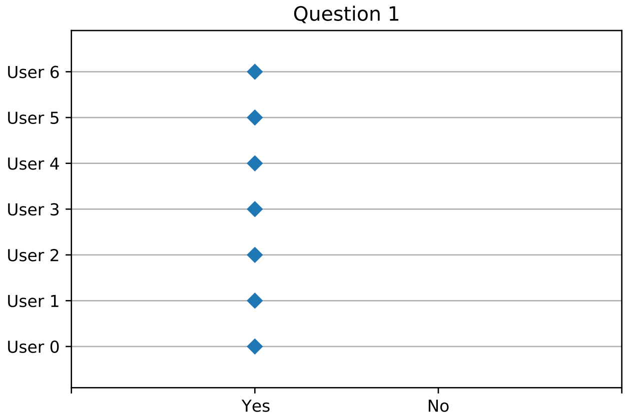


Figure B.1: Question 1: Do you have a formal education in music?

B.1.2 What are the genres and/or styles of your interest?

	Question 2
User 0	Classical
User 1	Classical, Rock, Rap
User 2	Electronic music, dubstep, dub, ambient, techno, house, electroacoustic music
User 3	Indierock, contemporary, electronic (early electronic), dreampop
User 4	Rock and jazz
User 5	jazz, contemporary music, krautrock, techno, drum n bass, classical, rock, rap, electroacoustic, etc.
User 6	I work on a lot of videogame music, and I like listening to various kinds of rock.

Table B.1: What are the genres and/or styles of your interest?

B.1.3 How would you rate your music improvisation skills?

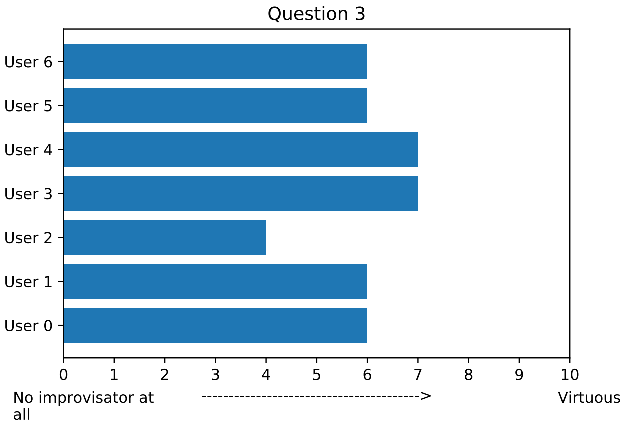


Figure B.2: Question 3: How would you rate your music improvisation skills?

B.1.4 Do you usually play with other musicians (band or others)?

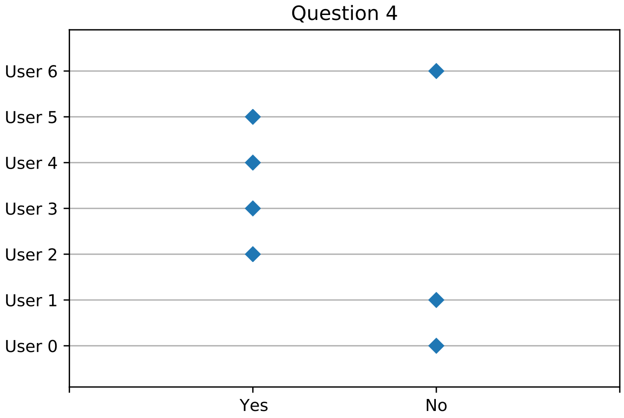


Figure B.3: Question 4: Do you usually play with other musicians (band or others)?

B.1.5 Have you use a looper device (digital or analog) before?

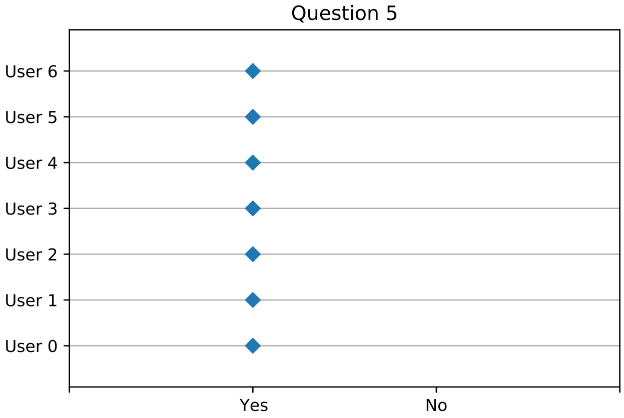


Figure B.4: Question 5: Have you use a looper device (digital or analog) before?

B.1.6 If your answer to the previous looper question was positive, what device or tool have you used?

	Question 6
User 0	A guitar looper
User 1	Loop Pedal, DAW Looping
User 2	Ableton Live
User 3	Hologram Microcosm, TC Electronic Alter Ego, Boss RC30 ++ Also different looper apps (iPhone)
User 4	Looper pedals for guitar, looping in Logic and Live
User 5	looper pedal, ableton, delay pedal.
User 6	Fruity Loops Studio

Table B.2: If your answer to the previous looper question was positive, what device or tool have you used?

B.1.7 How would you rate your keyboard playing skills?

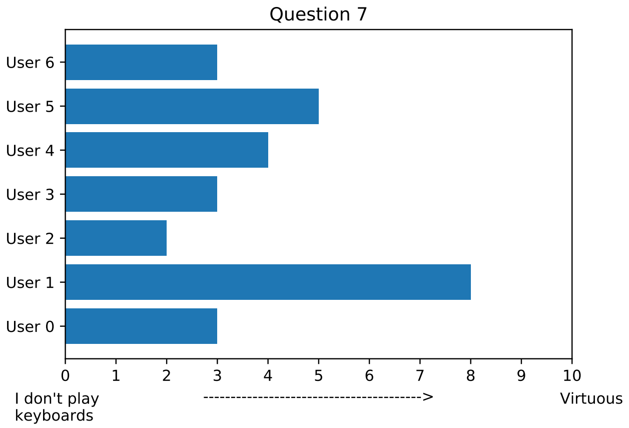


Figure B.5: Question 7: How would you rate your keyboard playing skills?

B.1.8 What is your experience playing sound synthesizers?

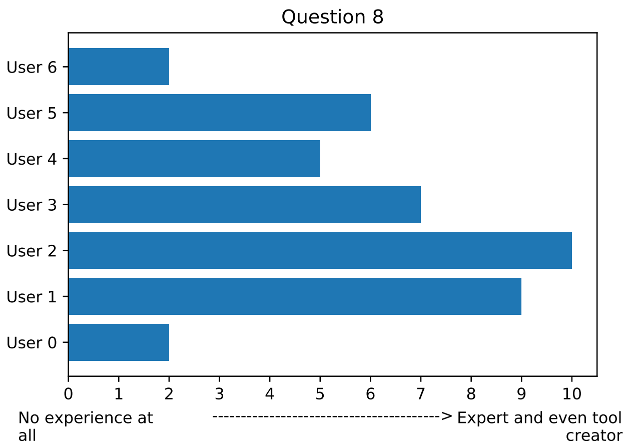


Figure B.6: Question 8: What is your experience playing sound synthesizers?

B.1.9 Explain briefly the kind of music that you usually compose/perform.

	Question 9
User 0	Classical
User 1	Classical Pieces, Electronic programmed in Pure Data/Max/ect.
User 2	Rhytmic electronic music with a focus on texture and sound design, but not really dance-oriented
User 3	Experimental electronic, and droning indie
User 4	Improvisatory rock
User 5	Compose mainly contemporary music and jazz. Play weird rockjazz.
User 6	I compose a lot of music primarily for videogames, which means I work with instrumental music in various genres. Mostly I've done rock, techno, metal, orchestral, performance and hip hop. Recently, I've also gotten more experience in composing, writing, performing and mixing in vocals into my work as well.

Table B.3: Explain briefly the kind of music that you usually compose/perform.

B.2 Extended Reality (XR) Technology Experience

B.2.1 Have you had experience with artificial intelligence for composing/performing music? if so, Can you briefly explain in what way?

	Question 10
User 0	No
User 1	Master level course in machine learning for music, creating a gesture triggered sampler using a variational autoencode
User 2	No
User 3	No
User 4	No
User 5	not really, but looking into rave.
User 6	Very minorly. I've occasionally used FL Studio's riff generating feature, and adapted that into riffs and chords to use in my music.

Table B.4: Have you had experience with artificial intelligence for composing/performing music? if so, Can you briefly explain in what way?

B.2.2 What of the following technologies have you use before? (You can choose more than one)

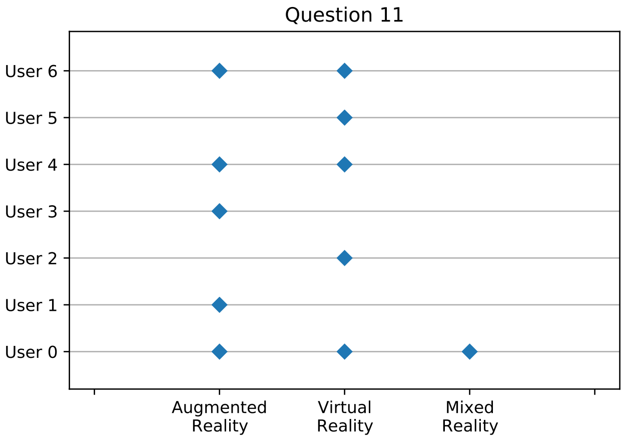


Figure B.7: Question 11: What of the following technologies have you use before? (You can choose more than one)

B.3 Music Session Experience

B.3.1 Have you use the Microsoft HoloLens headset before?

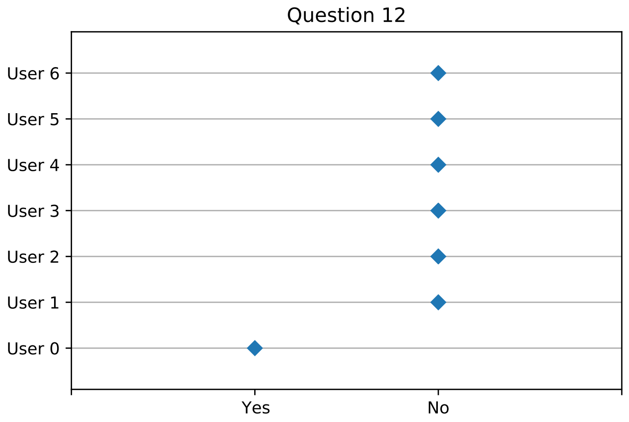


Figure B.8: Question 12: Have you use the Microsoft HoloLens headset before?

B.3.2 Fluency for playing: Did you feel a significant amount of latency (a delay time between the moment you play a key and when you hear the generated sound) when performing? (NEGATIVE question)

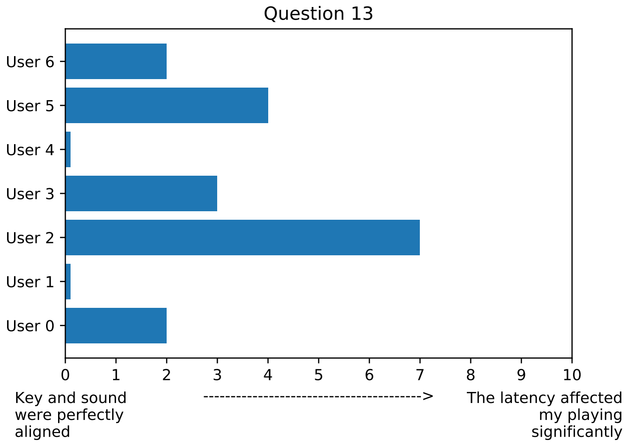


Figure B.9: Question 13: Fluency for playing: Did you feel a significant amount of latency (a delay time between the moment you play a key and when you hear the generated sound) when performing? (NEGATIVE question)

B.3.3 Positioning: (Just in terms of SOUND - from the speakers array) When you move the rigid body (object tracked by the motion capture system), did you feel (hear) that the sound was aligned with the physical movement of the object? (while moving it)

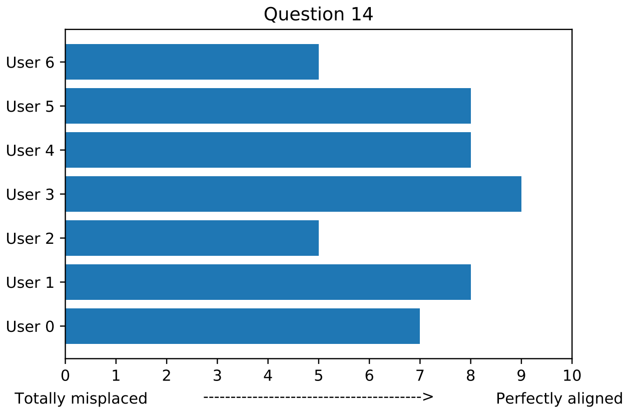


Figure B.10: Question 14: Positioning: (Just in terms of SOUND - from the speakers array) When you move the rigid body (object tracked by the motion capture system), did you feel (hear) that the sound was aligned with the physical movement of the object? (while moving it)

B.3.4 Positioning: (Just in terms of IMAGE - from the HoloLens) When you move the rigid body (object tracked by the motion capture system), did you feel (see) that the image was aligned with the physical movement of the object? (while moving it)

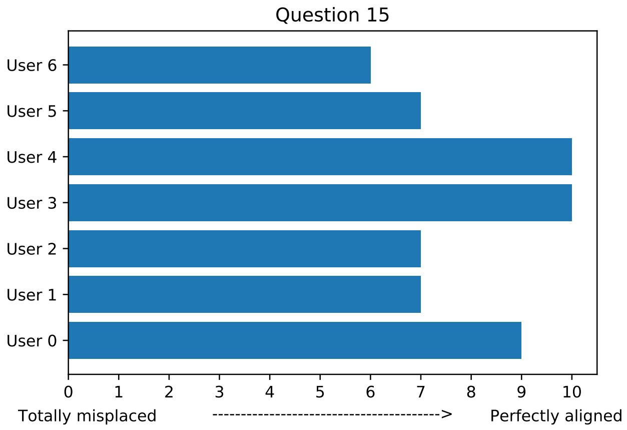


Figure B.11: Question 15: Positioning: (Just in terms of IMAGE - from the HoloLens) When you move the rigid body (object tracked by the motion capture system), did you feel (see) that the image was aligned with the physical movement of the object? (while moving it)

B.3.5 Sound location: How easy was to identify (just in terms of SOUNDS from the loudspeakers array) the location of your sound source (sphere) when it MOVED FREELY ?

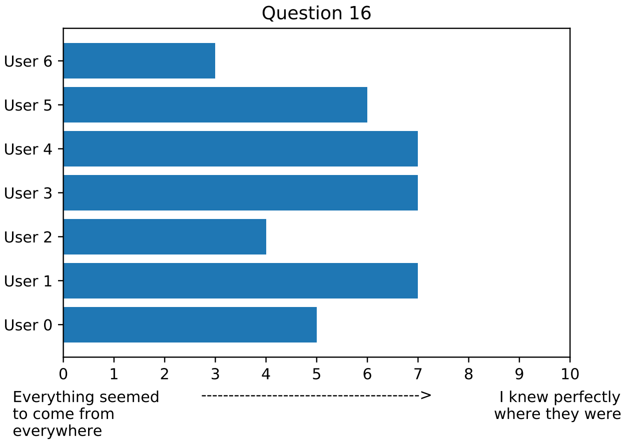


Figure B.12: Question 16: Sound location: How easy was to identify (just in terms of SOUNDS from the loudspeakers array) the location of your sound source (sphere) when it MOVED FREELY ?

B.3.6 Visual confirmation: Did you feel that the visual sound source (sphere in HoloLens) was aligned with the sound direction when MOVED FREELY? (e.g. when you were seeing one sphere going to your right side, was the sound following it behind or at the same time?)

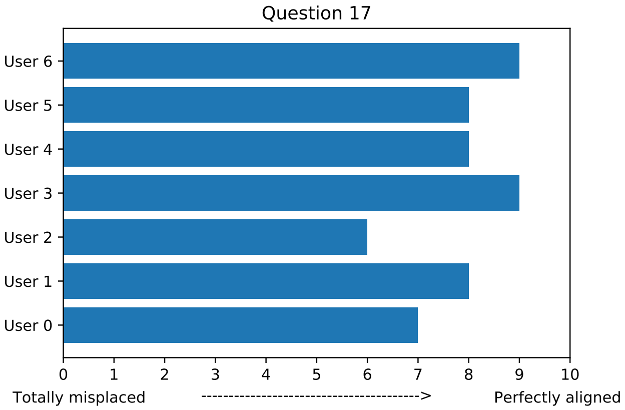


Figure B.13: Question 17: Visual confirmation: Did you feel that the visual sound source (sphere in HoloLens) was aligned with the sound direction when MOVED FREELY? (e.g. when you were seeing one sphere going to your right side, was the sound following it behind or at the same time?)

B.3.7 Visual confirmation: If you didn't have a sphere in your field of view in the HoloLens (because it was probably in your side or behind), how easy was to identify its location and look for it?

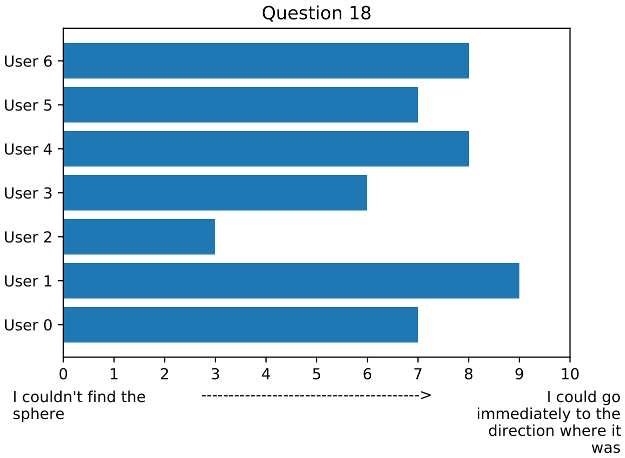


Figure B.14: Question 18: Visual confirmation: If you didn't have a sphere in your field of view in the HoloLens (because it was probably in your side or behind), how easy was to identify its location and look for it?

B.3.8 Agent Interaction: How easy was to ”release” (for autonomous movement) or ”catch” (to control with the rigid body) a sphere in the air through the HoloLens?

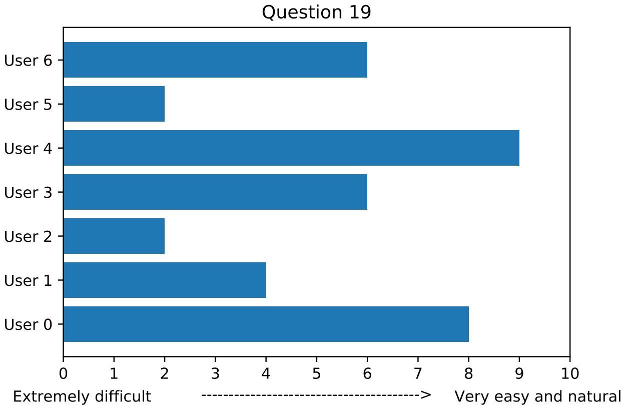


Figure B.15: Question 19: Agent Interaction: How easy was to ”release” (for autonomous movement) or ”catch” (to control with the rigid body) a sphere in the air through the HoloLens?

B.3.9 Automatic Movement: How did you perceived the speed of the spheres when they were moving autonomously?

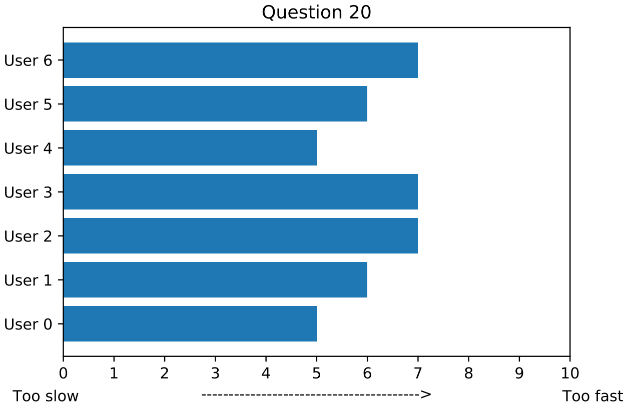


Figure B.16: Question 20: Automatic Movement: How did you perceived the speed of the spheres when they were moving autonomously?

B.3.10 Automatic Movement: Did you notice that spheres were, in some sense, following you?

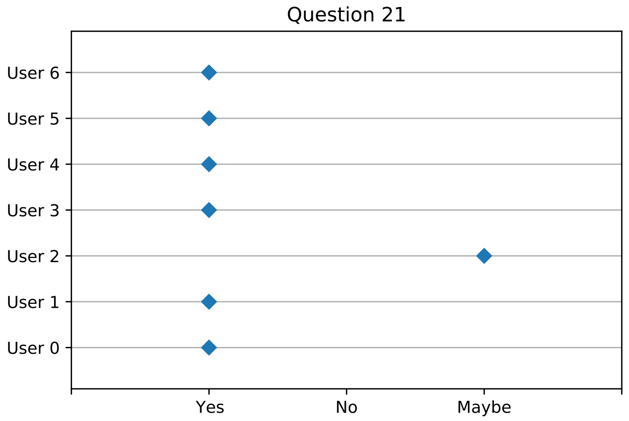


Figure B.17: Question 21: Automatic Movement: Did you notice that spheres were, in some sense, following you?

B.3.11 Automatic Movement: How accurate do you think you can predict the trajectory of one of the spheres?

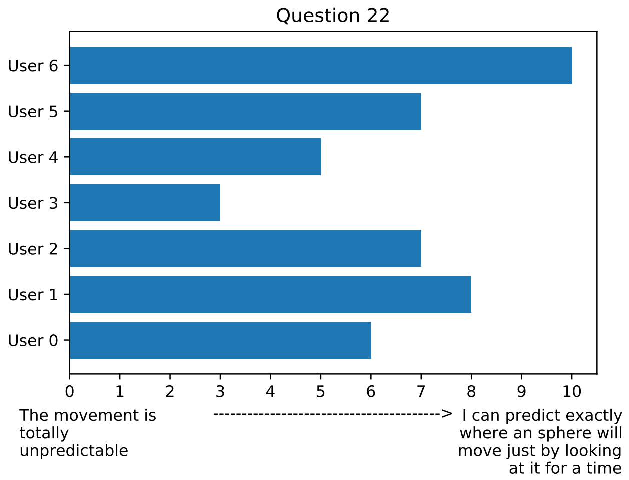


Figure B.18: Question 22: Automatic Movement: How accurate do you think you can predict the trajectory of one of the spheres?

B.3.12 Automatic Movement: To what extent (during the whole performance) you think that the spheres were spread equally around you or were grouped in specific directions when you look at them?

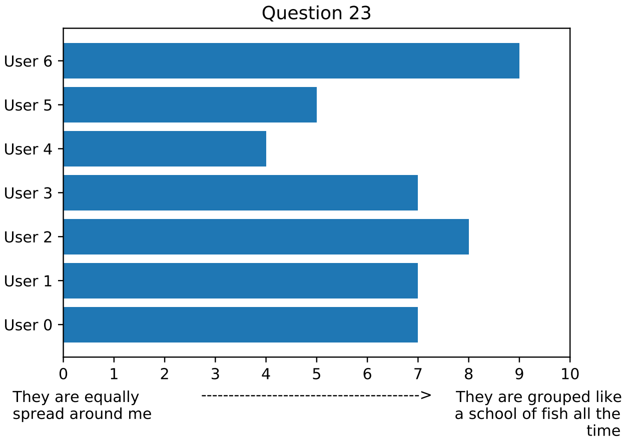


Figure B.19: Question 23: Automatic Movement: To what extent (during the whole performance) you think that the spheres were spread equally around you or were grouped in specific directions when you look at them?

B.3.13 Automatic Movement: According with your experience, describe the movement behavior of the spheres as a group.

	Question 24
User 0	They were going in different directions and with different speeds, but they tended to be a bit onesided of the whole "audio room". For instance; all of the spheres being behind me.
User 1	Like a cloud, following my movement
User 2	I felt like the spheres were often mostly on the same side as me, and often moved through me
User 3	I had a feeling they where following me, but hiding. Maybe they tried not to disturb me when I was working on one of them
User 4	Kind of like improvised dance, coordinated yet individuallistic
User 5	Flocking around me when standing still, running away from me sometimes, and coming towards me when i was moving backwards.
User 6	It was very much like a school of fish that circled around me, which was interesting to look and move around the room with. But I had some trouble at times with making out which sphere corresponded with a sound pattern I wished to edit because of it.

Table B.5: Automatic Movement: According with your experience, describe the movement behavior of the spheres as a group.

B.3.14 Looper usage: How easy was for you to use the looper in the MIDI controller?

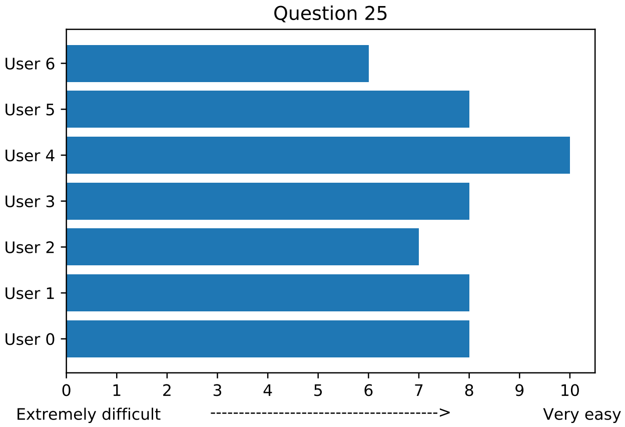


Figure B.20: Question 25: Looper usage: How easy was for you to use the looper in the MIDI controller?

B.3.15 Music Performance: Did you feel in total control of your music improvisation?

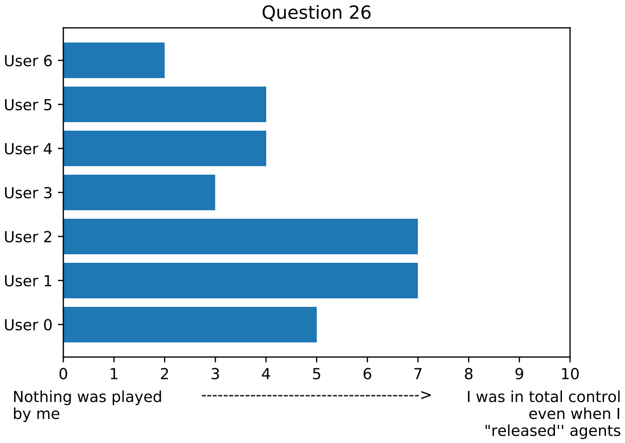


Figure B.21: Question 26: Music Performance: Did you feel in total control of your music improvisation?

B.3.16 Music Performance: Did you agree all the time with the musical material generated by the system?

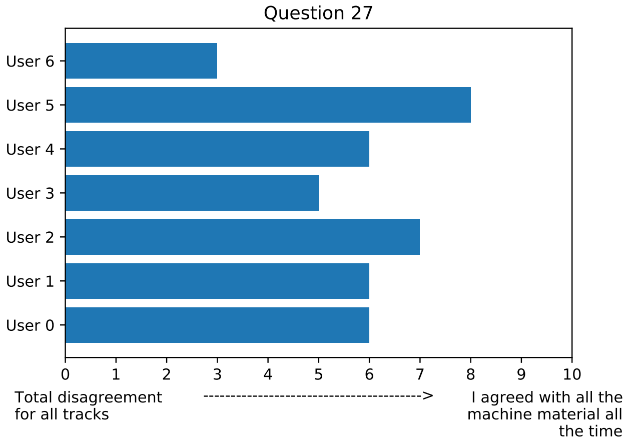


Figure B.22: Question 27: Music Performance: Did you agree all the time with the musical material generated by the system?

B.3.17 Music Performance: Do you think that the machine was close to your playing style and composition/improvisation vision of your piece?

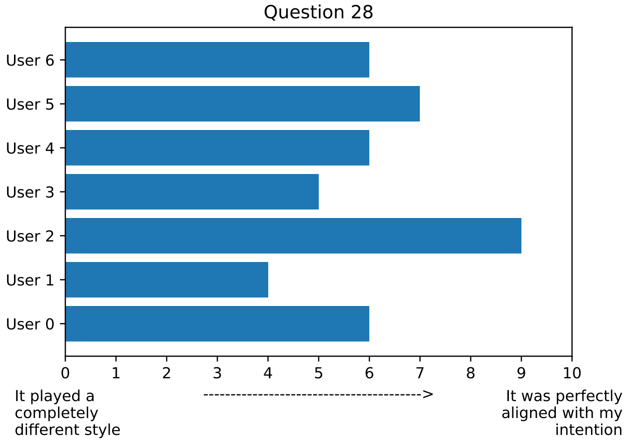


Figure B.23: Question 28: Music Performance: Do you think that the machine was close to your playing style and composition/improvisation vision of your piece?

B.3.18 Music performance: When the machine played its music, do you think it was in sync with the musical tempo (metronome)?

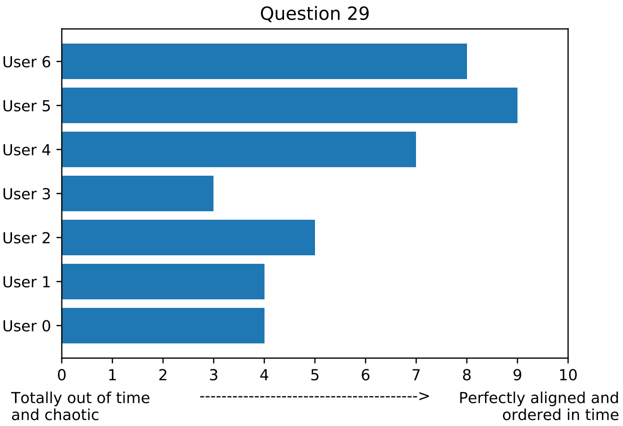


Figure B.24: Question 29: Music performance: When the machine played its music, do you think it was in sync with the musical tempo (metronome)?

B.3.19 Sound synthesis: How much did you change sound synthesis parameters (knobs in the keyboard) to find a sound that fits with your improvisation?

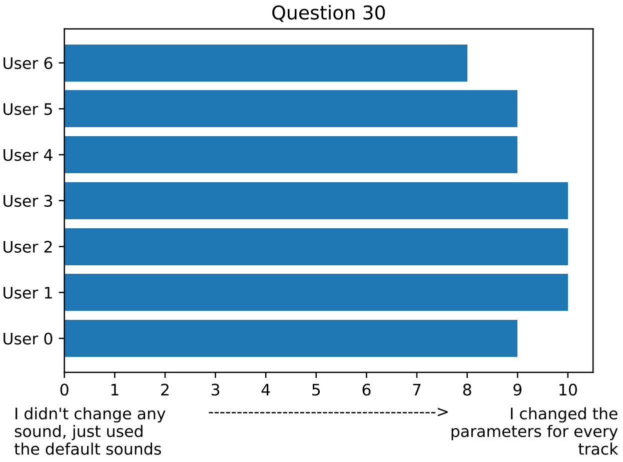


Figure B.25: Question 30: Sound synthesis: How much did you change sound synthesis parameters (knobs in the keyboard) to find a sound that fits with your improvisation?

B.3.20 Ergonomics: Did you feel the HoloLens headset comfortable during the whole session?

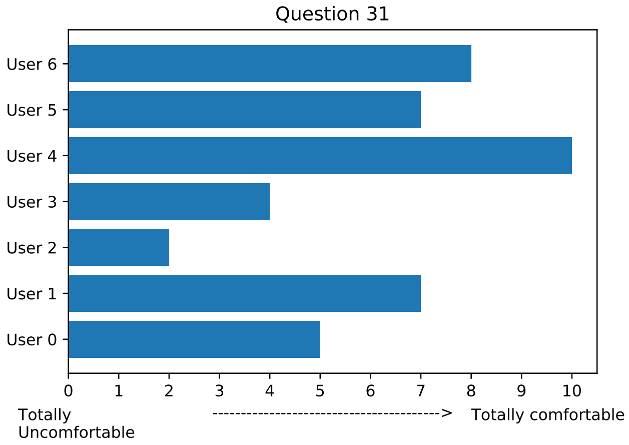


Figure B.26: Question 31: Ergonomics: Did you feel the HoloLens headset comfortable during the whole session?

B.3.21 Visualization: Did you feel limited by the ”field of view” (area of the mini-screen in the glasses) of the HoloLens while performing? (NEGATIVE question)

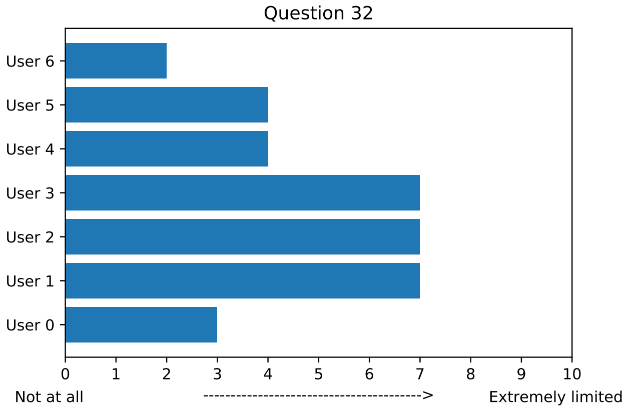


Figure B.27: Question 32: Visualization: Did you feel limited by the ”field of view” (area of the mini-screen in the glasses) of the HoloLens while performing? (NEGATIVE question)

B.3.22 Overall experience: Did you feel a well balanced sound mixing when you had several sound sources moving around the space?

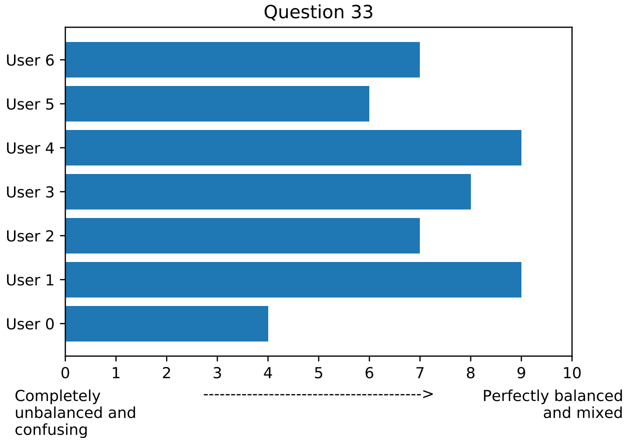


Figure B.28: Question 33: Overall experience: Did you feel a well balanced sound mixing when you had several sound sources moving around the space?

B.3.23 Overall experience: How many tracks/agents/spheres you think you could manage in this kind of system? and Why? (The system was up to 8 but you can suggest a lower or higher number that you think is right for you)

	Question 34
User 0	Hard to say, it depends on the lenght and complexity of the loops, but around 7-8 it was starting to become difficult to distinguish and keep track of the tracks/spheres
User 1	I think 8 was around a good number for the maximum. At 8, I started to slowly lose a bit of an overview of what each sound was doing, and was about the maximum of what I could remember doing for each, but this could also just be me as I didn't really plan out what I was going to do and just dived in making sounds.
User 2	I could have managed probably 15 or so if they were more spread and didn't move as fast/
User 3	Eight was enough. At around six the system also was a little stressed and the tempo was lagging/uneven sometimes. I used sounds not very dependent on syncronization, and that's also why eight tracks were managable for me.
User 4	8
User 5	a bit more than 8 with some experience
User 6	I went up to 8 spheres and had to stop myself once I hit that limit. Due to how the machine reinterpreted my inputs when the spheres were released, I started going over to creating a sort of soundscape, instead of a piece of music, and had fun experimenting with what sorts of additions I could add to it.

Table B.6: Overall experience: How many tracks/agents/spheres you think you could manage in this kind of system? and Why? (The system was up to 8 but you can suggest a lower or higher number that you think is right for you)

B.3.24 Overall experience: In general, how easy was to use the whole system?

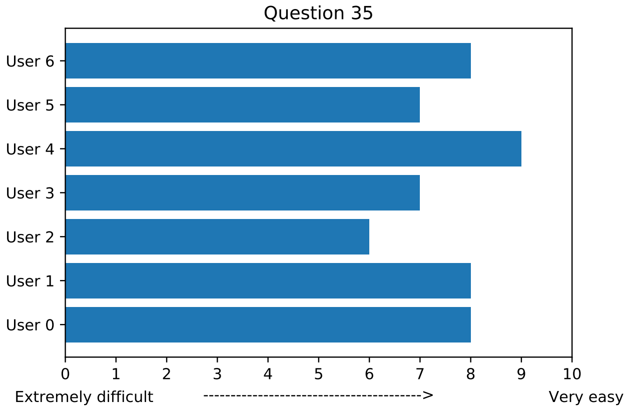


Figure B.29: Question 35: Overall experience: In general, how easy was to use the whole system?

B.3.25 Overall experience: How would you rate the AESTHETICS of the resulting music from your human-machine performance?

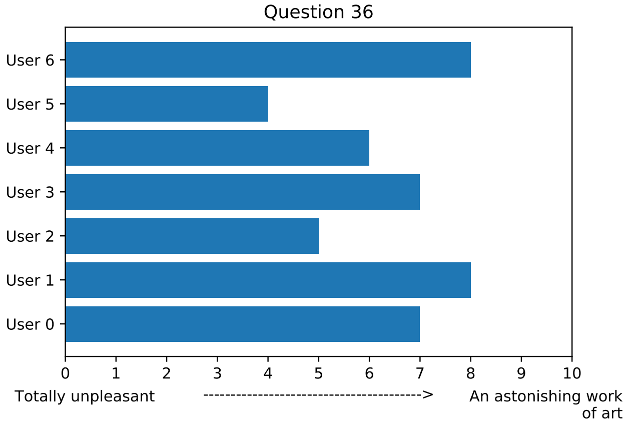


Figure B.30: Question 36: Overall experience: How would you rate the AESTHETICS of the resulting music from your human-machine performance?

B.3.26 Overall experience: How much did you enjoy the performance?

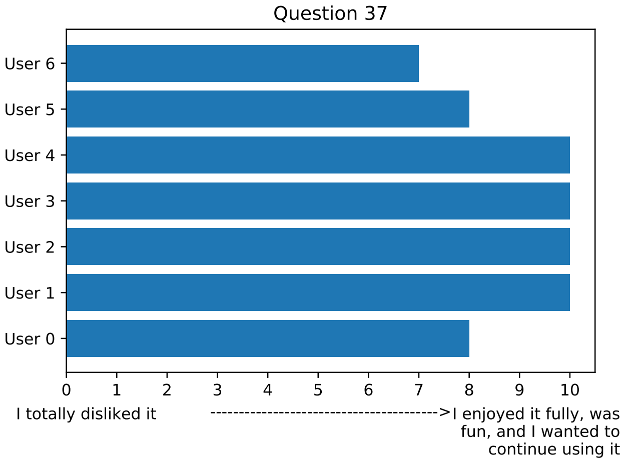


Figure B.31: Question 37: Overall experience: How much did you enjoy the performance?

B.3.27 Final question: What is your reflection, comments, suggestions and all your thinking about the system and your experience with it?

	Question 38
User 0	Some minor changes and improvements could be made, but overall it was pretty fun!
User 1	I thought the system was really fun and wanted to continue using it after it ended. It felt like interacting with another participant in the music and the overall aesthetic was really good. The only thing was that I sometime had problems 'clicking' on a sphere to return it but otherwise it was a really good system which I can see a lot of potential to to create music and explore with.
User 2	I thought it was neat and immersive and an interesting way to play with sound. I found that getting the looping timed right was difficult, and that it was impractical to make more conventional music. However, it was fun to do more ambient and electroacoustic things. I think the MIDI keyboard might not be the best controller, and I might prefer something a bit freer, maybe a gamepad controller.
User 3	It was a really fun way to play with sound. If I could have used this interface with my own hardware (synths, guitars and other), it would have been easier to create an astonishing piece of art. I would prefer to be able to control whether I'd have a metronome or not, and the looper should not be dependent on following it (i.e. it should not have to wait for the first beat to play back and such, but things like that should be an easy fix in future versions). The lenses are a bit heavy on my head, but after a while I almost forgot. It would also be great to have a mixer controlling all eight tracks also when released, for faster adjusting of sound rather than having to catch the sphere again. But all in all a really fun and enjoiable experience.
User 4	When catching a sphere all the spheres changed their numbering, this kind of made keeping track of the loops a bit confusing. Perhaps it could be an idea to name the spheres according to their synth presets?
User 5	Really fun! Could elevate the spheres a bit, and improve the catching-releasing mechanism. Would be nice to have more control over the syntheziser as well.
User 6	I definitely think it's a fun way to play with music creation. Due to how the machine changed how each sphere played what I had put in when I released them, I can't say I would use it much to create music tracks for listening purposes. But I would say this is a really interesting as something to introduce people into music creation. And also to experiment with and make musical ambience.

Table B.7: Final question: What is your reflection, comments, suggestions and all your thinking about the system and your experience with it?