UNIVERSITY OF LONDON
INTERNATIONAL PROGRAMMES

Going with instincts: Sensory Overload in Interaction Design

James Lanzon

Saturday 14th May, 2016

Supervisor: Dylan Seychell

Co-supervisor: Jonathan Barbara

Submitted as part of the requirements for the award of the Degree in **BSc Creative Computing** of the University of London.

Local Institution: St. Martin's Institute of Higher Education

SRN: 130428385

1 CONTENTS

2 ABSTRACT

Over stimulation of senses has proven to impact the cognitive and physiological state of individuals. As we find ourselves exposed to information continuously, the need for intelligent agents increases. This project explored the impact of sensory overload based on gathered qualitative and quantitative data. The former was collected based on participants' subjective experience in the form of a questionnaire, while the latter was collected using a combination of a cognitive test and the storage of the physiological inputs. Heart rate and galvanic skin response were used as dynamic physiological inputs to the system alongside a series of static musical compositions. Using fuzzy logic, the inputs were mapped to a variety of components within the implemented visualization. The user experienced the aforementioned mappings through either a traditional or a head mounted display. Analysis of the collected data was indicative that regardless of the display environment used, individuals appear to exhibit similar characteristics.

Keywords: *Interaction design, human-computer interaction, virtual reality, galvanic skin response, heart rate, sensory overload, fuzzy logic, head-mounted display, traditional-display environment*

3 ACKNOWLEDGEMENTS

I would like to start off by thanking my supervisor Dylan Seychell and co-supervisor Jonathan Barbara, whom have been great for their on-going support and for providing me the initiative to progress and further explore opportunities.

In particular, I would like to thank Silvio McGurk for his patience and much needed help at times hard to deal with. Special thanks goes out to all the staff at St. Martin's Institute of IT for all their guidance, help and patience. Furthermore, I would like to thank Dylan Busuttil and Kris Camilleri for without their outstanding help, this project would have been difficult to complete.

I would also like to thank Michael Camilleri and Daniel Xuereb for their help and all the participants that dedicated their time and patience to contribute to my research. Every participant was of great benefit and their interest motivated me to finalise my project.

Last but not least, I would like to thank my parents and brother for their patience when times got tough, and for being there through thick and thin. Your support was invaluable throughout this degree, without you all this would not have been possible.

4 INTRODUCTION

4.1 MOTIVATIONS

The main inspiration behind this project relates to the study of skin orgasms or goose-bumps that an individual experiences when listening to music they appreciate. This led to a growing fascination into the notion of sensory overload which served as the foundation of this project. Several subjects provided by the degree programme come in play: Creative Computing, Object-Oriented and Graphical-Oriented Programming in Java, Software Engineering, Interaction Design, Sound and Music, and Artificial Intelligence.

Aided by the programming principles and various techniques explored throughout the years, I attempted to explore another field in computing (Fuzzy Logic) which is not essentially part of the syllabus. The Creative Computing subjects (CO1112, CO2227) alongside Sound and Music also offered a variety of opportunities within the creative domain. They have served well for instigating creativity, desirable solutions and holding a respectable position in creating and criticising artefacts.

4.2 AIMS & OBJECTIVES

Looking back at the Preliminary Project Report, the aims and objectives have been narrowed down and are focused on the study of sensory overload in interaction design. In conclusion, I aim to use music and human-input in order to further explore sensory overload and how it could be used to improve the interaction of machines, interfaces and environments.

In order to achieve the aims stated above, the following objectives have been defined:

- *Research the impact of over stimulation of senses on productivity and cognitive performance presented in Sections 5, 7 and 8*
- *Use Fuzzy Logic to effectively map inputs to outputs in order to generate stimuli and examine the effect on individuals presented in Section 5*
- *To examine the effects of over stimulation on Heart Rate and Galvanic Skin Response of individuals presented in Sections 7 and 8*
- *To research the correlation between sensory overload and the impact it has using different display environments such as Head-Mounted Display (**HMD**) and Traditional Display Environment (**TDE**) presented in Section 8*

4.3 DELIVERABLES

- *GSR & HR analysers to allow input of human-response*
- *Fuzzy Logic Controller to create a visualisation taking as input human-response, frequency and amplitude*
- *Visualisation based on human and sound characteristics that works exactly in the same manner in both **HMD** environment and **TDE** with the exception for restricted screen rotation in the **TDE***
- *Questionnaires to compare exhibited data and cognitive experiences*
- *Experiments conducted on two separate groups, one experiencing the HMD and the other TDE*
- *Comparisons of gathered data related to user experience in form of results section*

This project requires a literature review which cover a variety of topics such as virtual reality, intelligent user interfaces, fuzzy logic and sensory overload. A system shall be developed to not only provide a medium for measuring human inputs but also to act as a feedback loop.

The system will gather human input and process it using fuzzy logic in order to map from one domain to another. Two environments will be designed to compare the HMD and TDE when individuals experiencing multiple stimulus simultaneously. Participants' cognitive and physiological performance will be measured before and after the experiment to measure any changes. The experiment will involve a visualisation that will display the human and musical composition inputs as a means to over stimulate the participants' senses.

Alongside this process, the cognitive performance will be evaluated against two short tests as well as gathering qualitative data from participants' response to the experience. The cognitive, qualitative and quantitative data will be used to further meet the objectives defined.

The results obtained from this process will be analysed in Section 7, discussed in Section 8 and further evaluated in Section 11 of this document as described in the aforementioned objectives.

5 LITERATURE REVIEW

5.1 INTRODUCTION

An **Intelligent User Interface** (*IUI*) which is a subfield of Human-Computer Interaction (HCI) **(Ehlert, 2003, p. 3)**, partially refers to interfaces that respond according to some level of artificial intelligence. An example of such an *IUI* is *Clippit* or *Clippy* of Microsoft Office (Meyer, 2016) which acts as a personal assistant.
Such agents should operate in the background or back-end of processes to allow users to perceive the application acting as an assistant. The user will fail to find use in an agent which performs actions that could have not been done by users, thus autonomy is essential for users to feel the effect of its assistance (Lieberman, 2016).

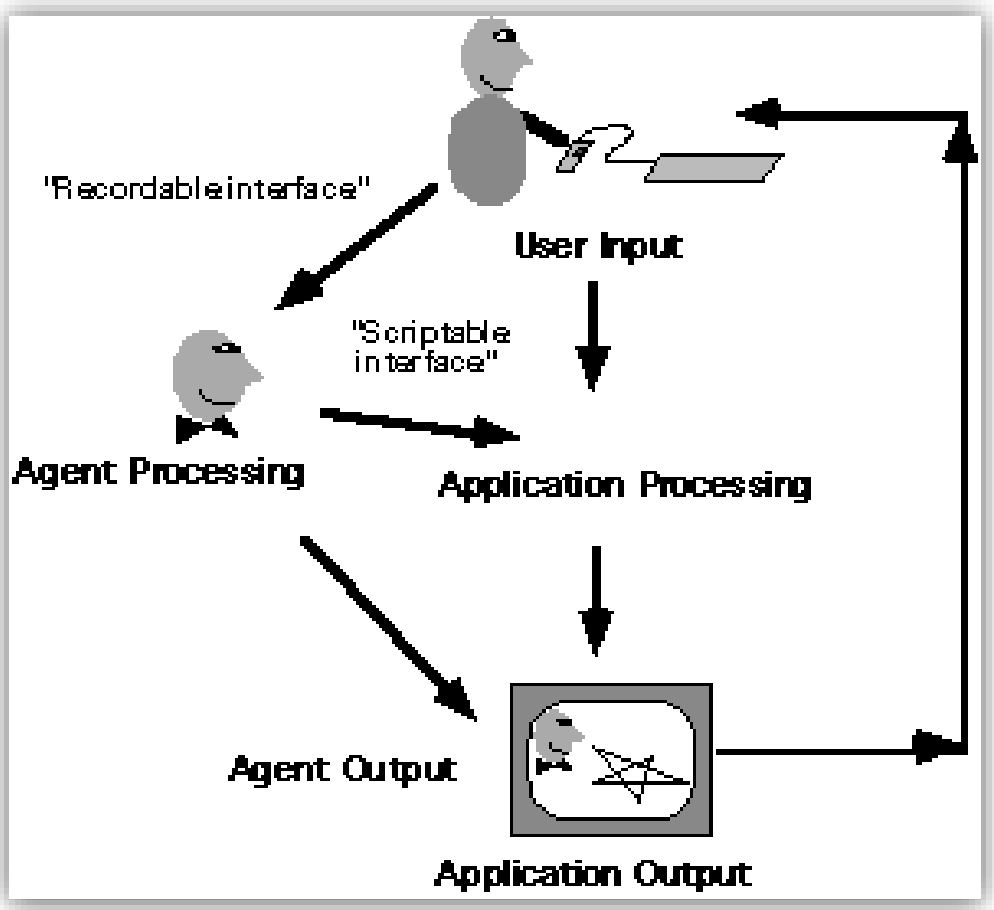


Figure 1**: Structure of an intelligent agent showing the transition from input to output **(Lieberman, 2016).

Since the agent operates in the background, it could observe user interface actions and respond by modifying the interface according to the action and conditions set by the user. There are also other types of agents which operate outside the user interface such as e-mail agents. Thus, for an agent to be classified as both an *IUI* and autonomous, it must follow that this agent should be part of the interface so that the agent operates autonomously in the background (Lieberman, 2016).

5.2 HUMAN-COMPUTER INTERACTION

5.2.1 Intelligent User Interfaces (IUIs)

As the use of computer programs increases, so does the complexity of their interfaces and their structure. For example Microsoft Word offers a large set of options and settings but not everyone is capable of using them or even know that they exist. It would be very useful to have an interface that adapts to the user's actions depending on the user's physical and mental state (e.g. when stressed) - obtained by sensors or any other technological means **(Ehlert, 2003, p. 1)**.

IUIs attempt to solve some problems that current interfaces cannot, such as:

- Creating personalised systems
- Information overflow or filtered problems
- Providing help on using new and complex programs
- Taking over tasks from the user
- Other forms of interaction

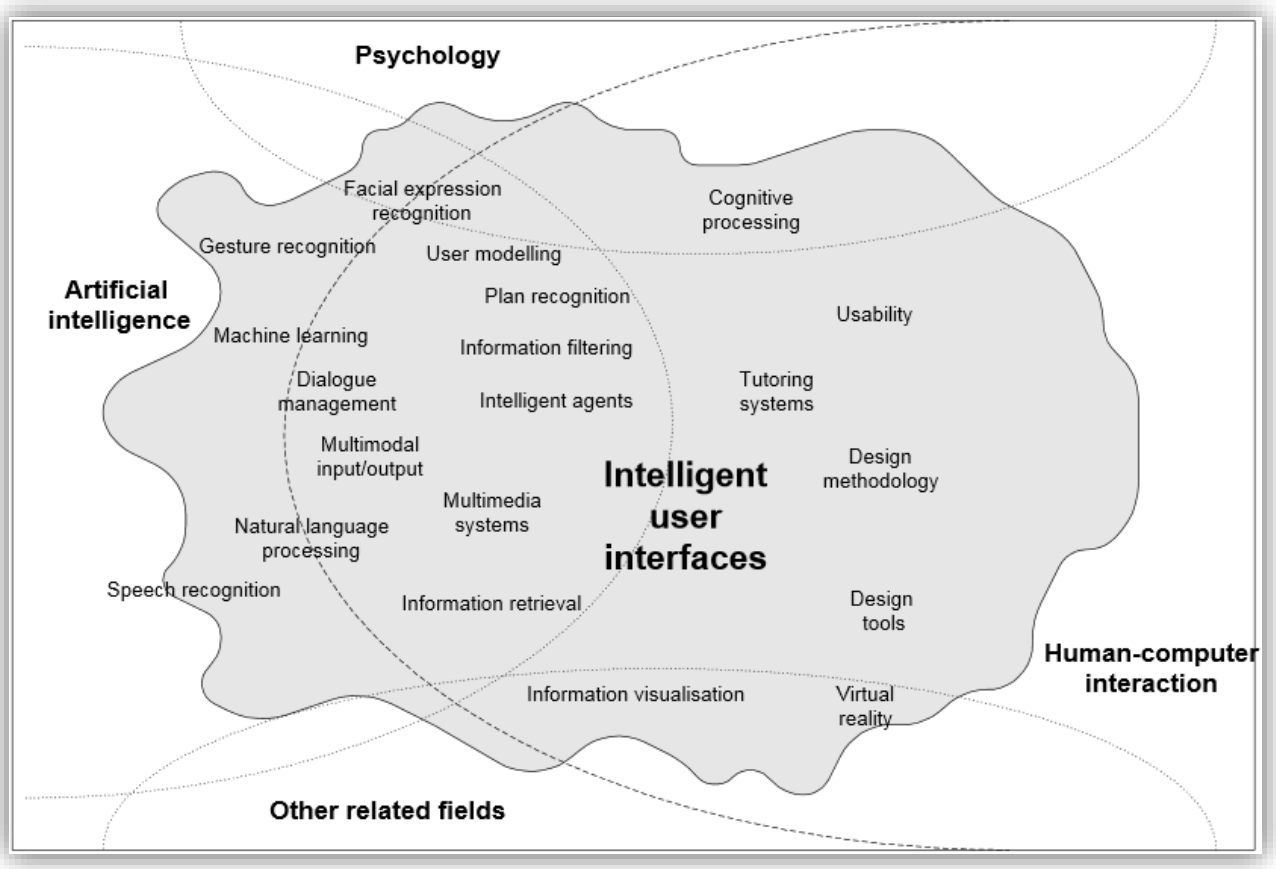


Figure 2: *IUIs research fields and relating topics (Ehlert, 2003, p. 13)*

By catering for such problems, the IUI could adapt to user actions by attempting to personalise the interface to match the state of the user whilst improving communication with them **(Ehlert, 2003, p. 2)**. There are other research fields that

helped shape IUIs, some of which include cognitive science, ergonomics, psychology and social sciences (see **Figure 2** for more research fields) **(Ehlert, 2003, p. 3)**.

Properties of an IUI:

- Intelligent input technology
- User modelling
- User adaptivity
- Explanation generation
- Personalisation
- Flexibility of use and so on

Traditional user interfaces are often created with pre-defined menus which are often an unproductive use of navigation (Khayut, Fabri, & Abukhana, 2014). These require the user to move from point A to point B without assuming difficulties. A lot of time can be wasted when using a poorly designed interface which in fact led to the study of usability principles and heuristics (Nielsen, 2016) and the general design of everyday things **(Norman, 2002)**.

Intelligent interfaces interact with users by using artificial intelligence, linguistics, rule inference technologies such as Prolog (SWI-Prolog, 2015), and many more which are used to enhance the experience **(Khayut et al., 2014, p. 1)**. These methods could improve the efficacy of human oriented tasks since these interfaces will adapt to the state of the user.

5.3 VIRTUAL REALITY

Virtual reality, virtual environment and augmented reality are all synonyms referring to the same scenario. However, in most cases many computer scientists and engineers prefer to use the latter two terms (Strickland, How Virtual Reality Works, 2016).

5.3.1 What is Virtual Reality?

The idea is simple; Virtual Reality **(VR)** replaces the reality you live in with some artificially generated environment that you or someone else might have created. This could range from roaming apps to games, simulations for military practice to avoid crashing actual machines and much more (Brownlee, 2016).

Computer technology is used to build a simulated three-dimensional world in which the environment may be programmed accordingly so that users may freely interact and change the environment, ignoring the real senses and feel immersed into that world.

Scientists, theorists and engineers have different opinions on what makes a VR experience but there are a couple guidelines to follow:

- Three-dimensional images should be scaled when in user point of view
- Sensor tracking is vital for users as the reflection of their movement onto the environment is what entails immersion (Strickland, How Virtual Reality Works, 2016).

5.3.2 **Stereoscopic display**

Stereoscopic display shows a different view to each eye in order to trick the eye into perceiving one whole image. This concept works the same as the way stereo headphones do when two different sounds are played in each ear in order to perceive one combined sound (Dr. Brian Jackson, 2016).

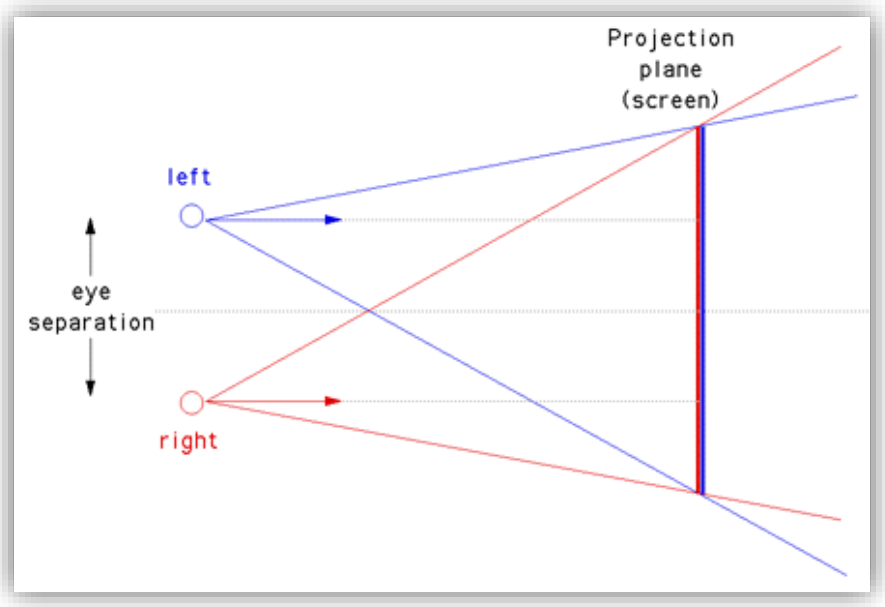


Figure 3: *Off-axis Stereo Pairs* ***(Bourke, 2016)***

5.3.3 **Head Tracking**

Without the appropriate means of tracking at least two of the five senses (sight and sound), immersion would be quite difficult to achieve (Rouse, 2016). Head tracking involves sensors that send data to the computer to indicate the shifting in the x-, y- and z-axis. Different companies have implemented various components to obtain better results and performance (Charara, 2016).

Such technology must ensure that users only experience low latency to be effective, otherwise the user might not feel any sense of immersion. In order to keep the user immersed in the experience, latency must be a thing of the past. Latency is the lag time from when the user acts to the moment the virtual environment reflects the action.

To further enhance the sense of immersion, sound is a key player in this scenario especially in the cases of 3D audio where the user would be tricked into thinking that the sound is coming from behind them or other places. This could be easily exploited in games, sensory experiments and so on.

Immersion will be further discussed however it is worth noting that it is imperative to the VR experience and also my research (as part of my objective is to compare the perceived effect from one group experiencing the experiment on a traditional screen to the other group experiencing the virtual environment) that the combination of audio and visual must go well together to enhance the sense of immersion.

How does it work?

Virtual Reality or Virtual Environments work as a combination of activities that must be dealt with once the user starts the interaction. For the user to be immersed the environment must capture certain senses and entertain with audio and video components.

Typical humans experience a minor delay when they turn their heads or movement of the eyes occur which has been researched to be around 50 milliseconds **(Strickland, 2016)**.

Interaction within the virtual environment is a must for users since it is a key element to engaging the perceptual senses of the user which can be achieved relatively easily with sensors and wearable devices, head-mounted displays (HMD), **(Strickland, 2016)** and other similar technology. The first HMD was created by Ivan Sutherland and his colleagues at MIT's Lincoln Laboratory known as the *Sword of Damocles* **(Rheingold, 2011, p. 3)**.

Figure 4: *A data suit to provide user input*
(Strickland, 2016)

3.1.1. Different hardware and viable options:

Many VR solutions appeared over the past few years with many interested in the potential of immersion when using a headset. Though one of the biggest issues that affects fully- and non-fully immersive experiences is cables (Humphries, 2016).

There are two terms that need to be discussed: *Virtual Reality* and *Augmented Reality*. The difference is quite obvious, *virtual reality* replaces your reality completely by some artificially programmed world whereas an *augmented reality* adds onto the reality you live in, a simple example is the Google Glass which has stopped its productions (Burns, 2016).

Augmented reality adds on graphics and sounds such as a GPS at the top right corner of your glasses to retrieve instant directions without obstructing your vision. There are many industries that could benefit from this e.g. tourists, soldiers and less-accessible which use computer-generated vision to aid them in their endeavours (Burns, 2016).

There are also two types of headsets which offer somewhat different products. The first type involves headsets that use smartphones as an engine like Google Cardboard or Samsung Gear VR. The second type involves standalone engines on Head-Mounted Displays (HMD) like Oculus Rift, HTC Vive which have gyroscopes and displays built-in. These help track movement and provide stereoscopic vision (possibly not great for the eyesight (Crawley, 2016)) to help the user become more immersed (Brownlee, 2016).

Other emerging display technologies involve eclipse LED-based virtual reality (Dr. Brian Jackson, 2016). Microsoft HoloLens and Magic leap used *laser waveguides* and *holography* for multi-focus 3D displays. This seems to make AR and VR experiences more natural to the eye.

Figure 5: *Virtual Reality Headsets* ***(Malmlund, 2016)***

3.1.2. Google Cardboard

VR headsets like Google Cardboard and Samsung Gear VR allow one to experience virtual reality without being restricted to the physical limitations of cable length. On the other hand, the experience is limited to the performance that the mobile device offers which could imply lesser immersive experiences though may still suffice for that situation. For best experience in terms of immersion and resolution quality, one should use headsets such as Oculus Rift, HTC Vive and so on (Humphries, 2016).

Google Cardboard is a simple VR headset that is cheap, portable and compatible with both android and iPhone devices. It is composed of nothing more than a piece of cardboard with two lenses and a magnet that costs less than $30 to purchase. The experience with this headset is not as immersive as wired headsets since Google Cardboard doesn't have a head strap and is designed to be held up to the face by your own hands (Miyao, 2016). The advantage this provides is that one could easily create a VR experience without the need to spend much.

The best part of the headset is the magnet which is an ingenious design aspect of Google cardboard (Ripton, 2016). It essentially acts as a medium to interface with the device without physically touching the screen. Every mobile device has a magnetometer that is used for compass functions. Once the magnet is moved/pulled down, a shift in the magnetic field is detected by the phone's magnetic sensor which in turn tricks the device into considering the shift as a click on screen (Sharma, 2016).

Figure 6: *Google Cardboard VR headset*
(Spradlin, 2016)

3.1.3. Immersion

Immersion of virtual environments is dictated by two primary characteristics which are essential to the experience. The first characteristic associates itself to the display mediums such as full HD screens or full HD goggles while the latter characteristic is based on the human-computer interaction methods **(Rheingold, 2011, p. 2)**.

It is what makes the user feel as being inside or part of the world or environment (Strickland, How Virtual Reality Works, 2016). A computer scientist named Jonathan Steuer proposed two main components of immersion: *depth of information* and *breadth of information* **(Steuer, 1992)**.

There are three types of immersive systems as explained by **(Kalawsky, 2000, p. 2).** A fully-immersive system is a display allowing 360° world space and control, semi-immersive systems are those that show less than 360°, and non-immersive systems are typically reserved for desktop VR systems.

Steuer called the combination of immersion and interaction with the environment *telepresence*. The degree of control with regards to the interaction with the environment plays a key role in immersion combined with other factors like minimised latency, HD resolution and so on since the level of quality must be high to endure a better experience.

Immersion, which also applies for presence discussed in the next section, relates to the use of different perceptual senses being stimulated **(Rigby & Richard, 2011, p. 1)**. The only way to enhance the immersive experience is to improve the aspects of simulation which involve sound, computer display, colour schemes, interactive aspects and so on.

In applications of simulation, it is vital for the VR experience to be as fully-immersive as possible. Such systems provide a detailed computer generated model in a 3D space created by developers or scientists for other people to train on in cases where the standard practice might involve someone's life, risk danger or expensive equipment. It is in such cases that without the user being immersed with the system, then the training will fail to suffice for applications such as Flight Simulators, Surgery Simulators and so on **(Rheingold, 2011, p. 2)**.

Latency is an issue that pertains to both the computational and the human vision aspects which overall affects the state of immersion. Human vision perceives a temporal delay known as latency which is caused by an object being within the field of view of the observer and how it's perceived by said observer (Feng, Mech., Technol., & Huhhot, 2016). Similarly there is what is known as *Lag time* between the head movement and the resulting image when using a VR system. This lag has been sufficiently reduced to as low as 20 milliseconds however it is still one of the main issues facing VR displays **(Rheingold, 2011, p. 5),** (Zhang, 2016).

3.1.4. Telepresence

Telepresence is sometimes also referred to as presence, and is often used to refer to the sense of 'being-in' the computer generated environment (Virtual Reality/Environments). Such an effect may require various senses to be stimulated in order to feel as though they are present in that scenario. Also, prior experiences may lessen or heighten the effect of presence **(Kalawsky, 2000, p. 1)**.

There doesn't seem to be any common definition of presence since its measurements are based on user experiences which are completely subjective to both new and previous experiences. Some might also attempt to suppress the feeling of presence, especially those in suspension of disbelief **(Baños et al., 2004, p. 1), (Slater & Wilbur, 1997, p. 5)**.

Several authors agree that presence is divided into two categories of variables: media characteristics and user characteristics **(Baños et al., 2004, p. 1)**. Media characteristics are further divided into media form and media content. Media form involve any forms of display like full HD monitors and interaction with the environment while the content is the actual content like videos, animation scenes, graphics etc. for the user to see. On the other hand user characteristics are the general details with regards to their age, gender, previous experiences and so on.

The clarity level in which the media characteristics are experienced must be of highest quality. Such characteristics include full HD resolution, fiber-optic wires, clear and crisp sound, and so on. To be immersed with the system, the user must feel as though they are within the environment itself, tricking the brain into forgetting its actual existence outside the virtual environment **(Edwards, 2011, p. 1)**.

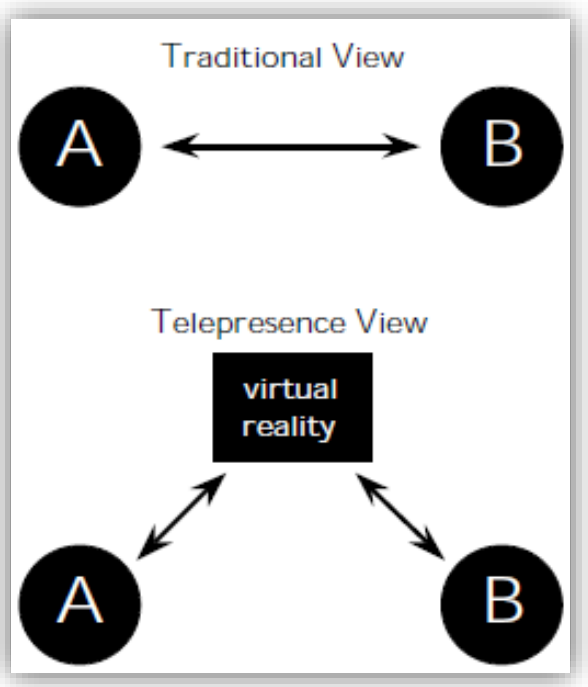


Figure 7: *Two models of mediated communication (Steuer, 1992, p. 8)*

Virtual presence can be communicated in various applications where the real worlds behaviour is mapped to the virtual space. Examples of such cases include a mobile personal assistant like Google Now, Microsoft Cortana or Apple Siri, community forums with personal avatars, Video Games like Sims 3 or World of Warcraft, and even virtual classrooms for a course (Wolf, 2016).

The key difference between presence and telepresence is that the first term refers to natural perception of an environment where the latter refers to mediated perception of an environment since telepresence is communicated using technology where presence refers to the immediate physical surroundings which sensory input directly affect the senses **(Steuer, 1992, p. 6)**.

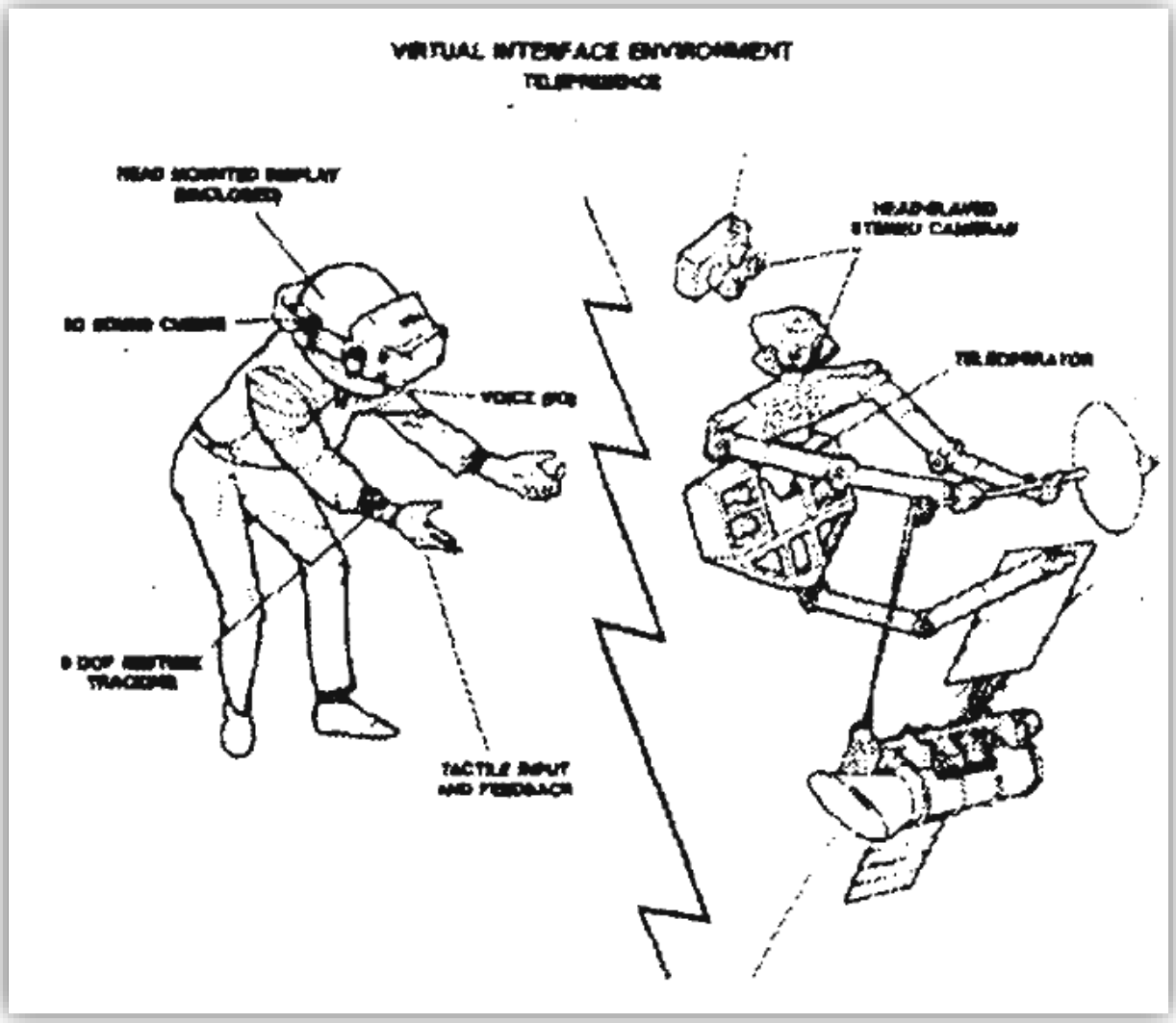


Figure 8: *The Virtual Interface Environment (Rheingold, 2011, p. 22)*

Carrie Heeter describes three dimensions of presence with regards to the experience of 'being there' **(Steuer, 1992, p. 7), (Heeter, 1992, p. 2)**:

- Subjective Personal Presence
- Social Presence
- Personal Presence

3.1.4.1. Subjective Personal Presence

This measures to what extent the user feels immersed in the world and the reasons of feeling part of the virtual world. There are countless reasons that contribute to this measure such as precise motion tracking, accurate depictions of scenery, learning environments and so on **(Heeter, 1992, p. 2)**.

3.1.4.2. Social Presence

Social and Environmental presence both hold the same arguments as to the extent in which other beings or the environment itself co-exists, reacts and also interacts with the user. Examples of social cues involve communication between one character and another like in 3D virtual worlds (SecondLife, 2016). In the case of environmental reactions, perhaps doors automatically open as you approach some proximity or trigger some trap as you walk in a game **(Heeter, 1992, p. 2)**.

3.1.4.3. Personal Presence

To create a powerful personal presence, the VR experience must be highly immersive by taking into consideration multiple sensors for accurate head tracking, position trackers, and interaction within the environment and so on. The feeling of being included in the virtual world is what enhances the feeling of personal presence. Seeing your hand move in the virtual world as you physically move helps convince the user as *"being there"* **(Heeter, 1992, p. 3)**.

To further shape our understanding of Virtual Reality without direct reference to different hardware options, a user perceives VR to act as a real environment which is truly simulated and thus experiences the sense of telepresence.

3.2. INFORMATION VISUALISATION

To see is to perceive. That is our brain works hard to make sense of all the stimuli reaching our senses every second. It is taken for granted how much of what we see, taste, feel hear and touch is mainly based on perception (Boundless, 2015).

We extract data to view, explore and understand information efficiently. Human perception can be divided into different stages as some authors prefer to segment the stages further however, in this case, perception consists of two stages (TheInnovationAcademy, 2015):

- Low-level visual properties are processed
  - Colour
  - Texture
  - Orientation
  - Movement

- Identifying objects by matching visual characteristics with properties stored in memory

On the other hand, another source splits the perception stages into three categories having the first two points refer to the first stage in the previous method above (Boundless, 2015):

- Selection
- Observation
- Interpretation

Figure 9: *Rubin's Vase – optical illusion showing the difference in perception of stimuli* ***(Boundless, 2015)***

Information visualisation is the use of computer supported interactive visual representations of data to amplify cognition. Using existing techniques, we can present

data to target audiences for them to derive conclusions easily (chibimaria, 2015). Proper visual representation of an interface can help users absorb larger amounts of information by condensing a lot of data into a simple graphic.

Typical traditional information visualisations tend to not describe any *design rationale* behind the authors choice of mappings, colour schemes, and the efficiency of information displayed. We assume that the author must know what works best for the visualisation and what does not, assuming they follow standard practices. It is typical to use correlations between human cognition and their respective visual perception as described on pg. 12 (Gestalt Laws of Psychology and Cognitive Science) **(Vande Moere, 2008, p. 2)**.

Interactive animation helps to exploit perceptual systems abilities to track relationships (TheInnovationAcademy, 2015) between moving objects. This follows upon the second stage suggested by (TheInnovationAcademy, 2015) which, for example, could take on a dynamic visualisation displaying the increase in pollution overtime as more illegal logging occurs (Deforestation) (Bradford, 2016).

The trend of Virtual Reality is bringing visualisation back to life as indicated by (Wheatley M. , 2016). DARPA had experimented with Oculus Rift back in 2014 which saw them eager to weaponise the Rift (Wheatley M. , 2016). This is of course a long time ago when considering the advancement in technology every year however one of the earliest pioneers in viewing Big Data using Virtual Reality is Dr. Creve Maples who has an undergraduate degree from MIT and a Ph.D in Nuclear Science from the University of

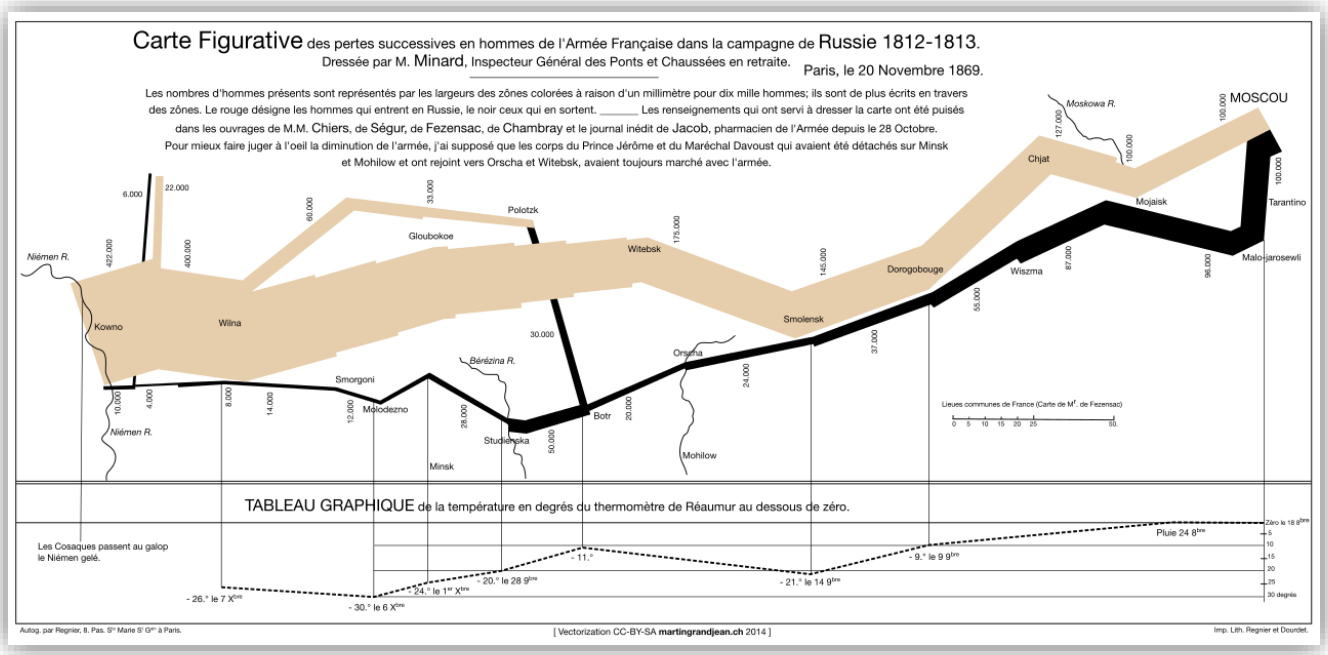


Figure 10: *Charles Minard's visualisation of Napoleon's March for 1812-1813 (Tufte, 2001, p. 41)*

California, Berkeley (O'Reilly, 2016), wanted to immerse himself in data. This led him to delve into fully immersible virtual reality environments to visualise data.

An interesting case study which involved a virtual environment as one of the breakthroughs in the race car industry was that of The Good Year Tire & Rubber Company struggling to figure out why they can't win races (Wheatley M. , 2016). Dr. Maples explains that using virtual environments to visualise data helps put you in places you

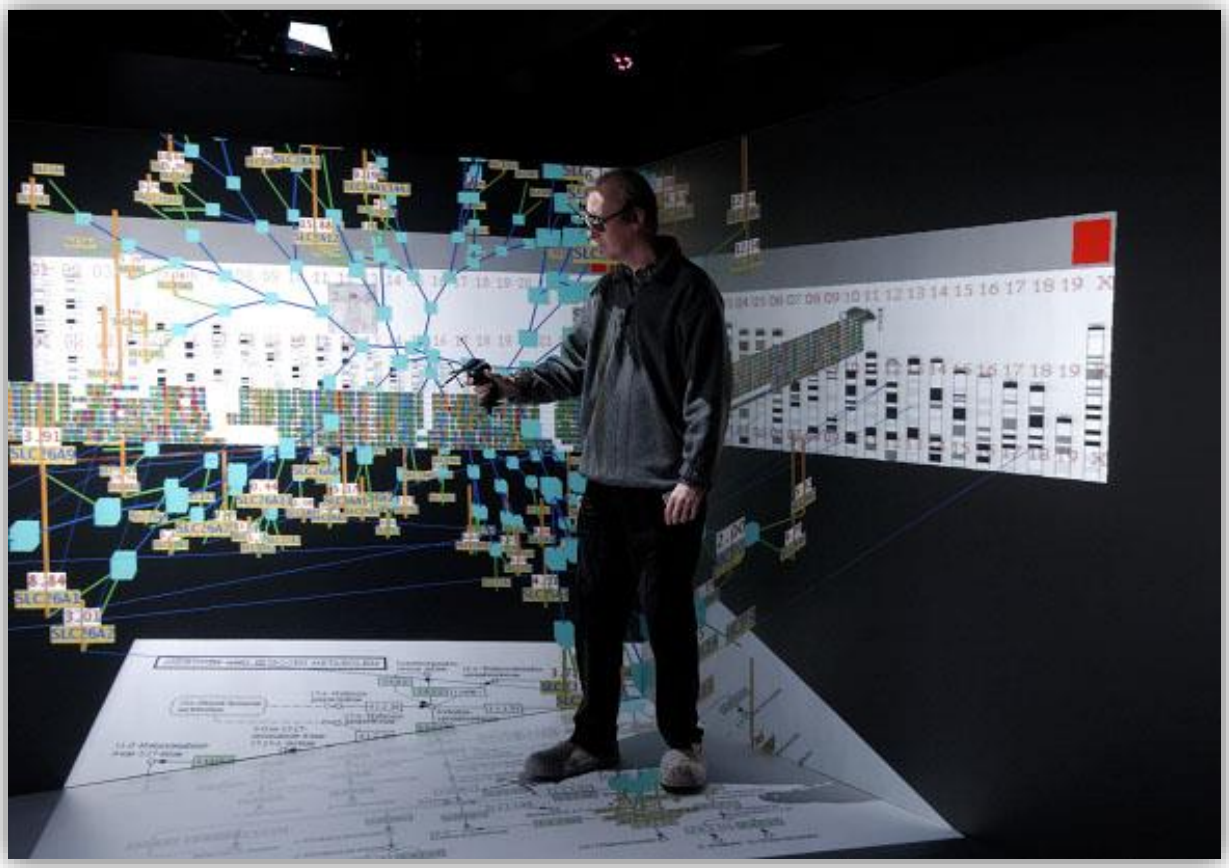


Figure 11: *Anton Koning mining the human genome in the new virtual reality environment* ***(Barco, 2016)***

can't physically go in real life. This will allow one to literally immerse themselves inside the environment that is the data. Dr. Maples then worked on a virtual model for two months which replicated the company's car for the engineers to use.

Everything that usually happens in a race car is exaggerated purposely to bring out extra effect on any possibility of something minor that could be keeping their cars back. Needless to say according to Wheatley, within five minutes, Goodyear's engineers found the answer to the problem. This case study motivates the community to further their research and experimentations in hope to not only contribute but also encourage the public how powerful information visualisation combined with virtual environments can prove to be.

> *"The key point is that "big data science" is not about data: it is about discovery and understanding of meaningful patterns hidden in the data."* ***(Donalek et al., 2014, p. 1)***

As **(Donalek et al., 2014, p. 5)** said in their paper, using technology such as immersive Virtual Reality will help maximise human pattern recognition and allow us to visualise data in a more intrinsic manner.

3.3. INFORMATION RETRIEVAL

Information retrieval is the process of obtaining and storing data to be used in the future. The huge advancement in computers and data processing techniques has allowed us to retrieve large amounts of data at breakneck speed, thus improving government and commercial infrastructure and the means of transporting and accessing their resources (Ishikawa, 2016).

The collection of data stored as records by abiding to certain structured rules is called a database. It essentially acts as a collection of files with the difference of how it is viewed, accessed, manipulated and secured. Information Retrieval is a combination of both text and image retrieval which forms part of Information Management (Nieuwenhuysen, 2016).

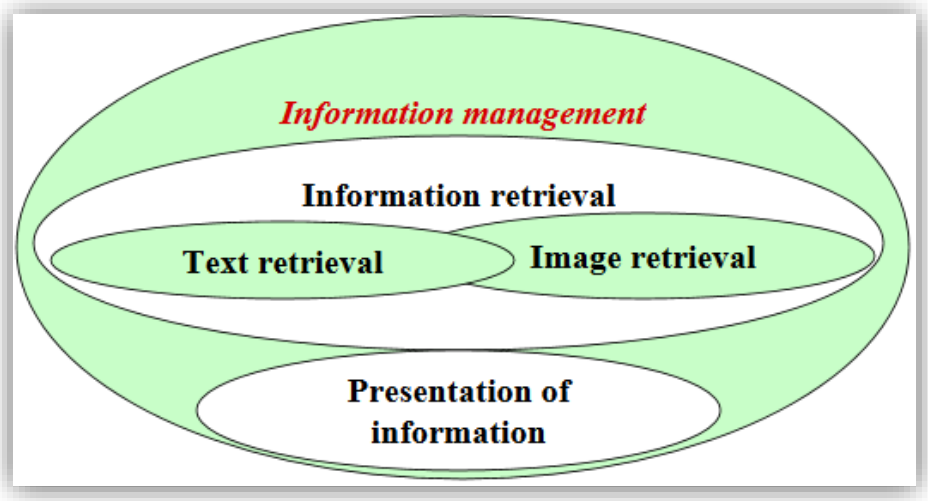


Figure 12: *Information Management set and its embedded subsets as shown by* ***(Nieuwenhuysen, 2016)***

Both *Text* and *Image* retrieval are considered to be part of *Information Management* due to the large overlap in both sections. In most cases, image retrieval is not completed using algorithms fetching real-time data from the image directly, but by using annotated images and any other text that has been used to define that specific image (Chastain, 2016)**; (Sergieh et al., 2012)**. These tags are then used in the search utility to sift through the database content and choose images that closely match these tags. This process is further enhanced with modern optimisation techniques such as indexing (Kennedy, 2016) as a way of storing and fetching data from a database more efficiently. Image retrieval is thus mostly based on text retrieval which uses keywords to identify particular properties in images.

Text retrieval applies certain techniques to fetch content with the use of keywords, or parts of a sentence (Stemming and Lemmatization (Stanford, 2016)) to facilitate a search query. Search utilities should ideally be designed to always return results however this sometimes may not be the case. By doing so, one could apply specific restrictions to direct the user selection when constructing their search query. This in turn allows a user to formulate a query such that any parameter selected should match something in the database.

There may be cases where users do not find matching filters/parameters for their required search, which could mean that their query is not suitable for that engine or database.

3.4. INFORMATION FILTERING

Information filtering (**IF**) is vital for the background process of Intelligent User Interfaces. Cognitive overload applications benefit from this aspect since the user could avoid reading all the information to perform decision-making. An intelligent adaptive system is used to filter through this information and display it in a more individualised or personalised manner **(Höök, 2000, p. 1)**.

Information filtering is a method of helping users do a variety of things. This includes the filtering of a wealth of information for the user to only focus on what relates to their interest (personalisation), support the use of a complex system and also being delegated tasks by the user to elevate the cognitive load of users **(Hanani et al., 2001, p. 1)**..

Typical filtering applications include search result filters for search engines, directories, e-mail directory, and archive of books and so many more. There is however a distinct difference between IF systems and information retrieval (**IR**) systems **(Hanani et al., 2001, p. 1)**. IF is essentially a subset of IR since both techniques have similar goals, which is to extract/select relevant information to the query **(Hanani et al., 2001, p. 3), (Belkin & Croft, 1992, p. 2)**. They differ in terms of how often each technique will be used.

In general, IR systems are technically designed to function once however also able to be used again (for e.g. a search filter in a database) whereas IF systems are designed to withstand long periods of use since the ultimate goal for IF systems is to assist the user in real-time.

5.4 PSYCHOLOGY

5.4.1 Cognitive Psychology

Cognitive Psychology is the study of mental processes involving how we think, perceive our environment or objects, shapes and colour, remember memories, people's names, recognise and also how we learn. Of course, this field is only a subset of the larger field of Cognitive Science (Cherry, 2016).

Between 1950 and 1970, behavioural psychology started to focus more on the fields relating to attention, memory and problem-solving. It was in this period that the term *Cognitive Psychology* emerged and it is often referred to as the cognitive revolution which took place at Harvard since there was much research put into these topics (Cherry, 2016) and has since included other topics as mentioned in the figure below.

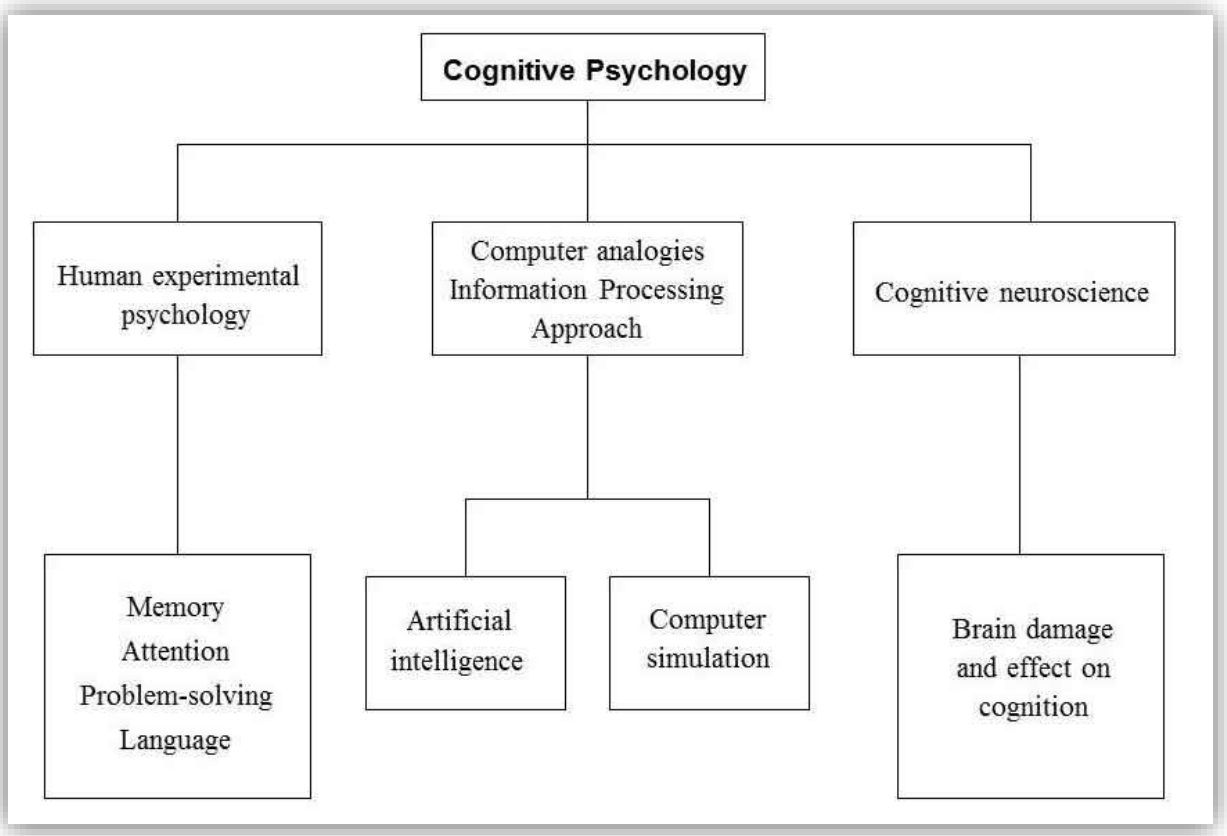


Figure 13: *Focuses on the way human process information* ***(McLeod, 2016)***

Cognitive psychologists have created new ways to study the human mind. By taking a similar approach to computers, creating many small units of knowledge to represent the current state of mind as information flows through the mind and senses of individuals. Some of this information can trigger two types of memory, immediate (working memory) or long-term memory (Cherry, 2016).

5.4.1.1 Types of Memory

Cognitive psychologists consider two main types of memories. Working memory refers to 'short-term' memory and is responsible for the immediate response to information sensed and typically, nothing tends to stick and is frequently forgotten by the next hour. The second type of memory is the long-term memory which is what we use to permanently store fascinating, never to-be forgotten memories.

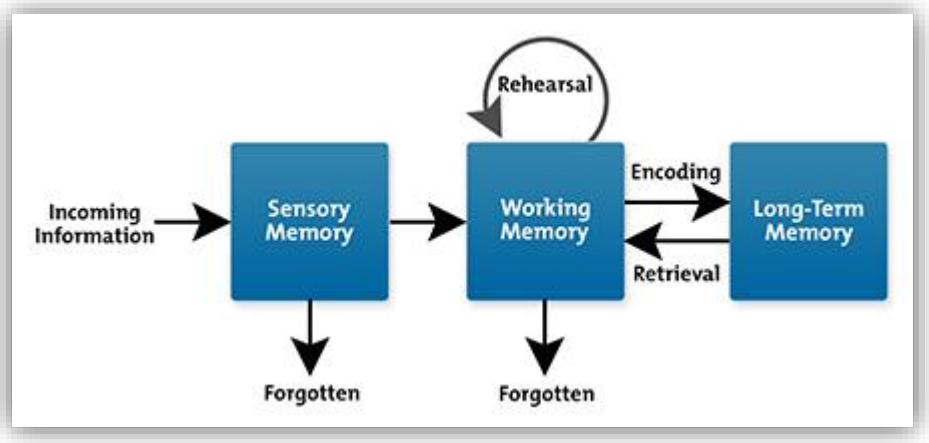


Figure 14**: The process of sensing information and the passage from sensory memory to the working memory onto long-term memory* ***(Mind Tools, 2016).

5.4.1.1.1 Working Memory

This is one of the three parts of memory which has already passed the stage of filtering through information (sensory memory) and currently processes information at the short-term or working memory stage. It is also known that this area holds information for a short period of time (approximately 20 to 60 seconds) (Diamond, 2016).

There are different opinions on how big a load one could hold in their working memory as they go about their activities. Educational psychologists approximate around two to three learning tasks simultaneously (Diamond, 2016) whilst other psychologists believe that the working memory is capable of holding between five to nine tasks of information at any time (Mind Tools, 2016).

For Example, when studying complex topics, taking notes at important times, paying attention, and studying the material will help lodge the information in the long term memory.

5.4.1.1.2 Long-term Memory

It is known that long-term memory holds information indefinitely and doesn't necessary have a limit on how much it could hold. Yet, the biggest challenge is finding information in this part of memory (Diamond, 2016).

Retrieval of memories from this part of the memory is vital to specific tasks however doesn't really impact our daily activities as most of that is stored in the working memory (Saul McLeod, 2016). Although one would require their long-term memory in order to complete tasks that they were oriented to do such as their job. It takes a certain amount of skill to perform a job and much of that is stored in long-term memory as one would have learned how to perform related tasks and studied to reach the position they are in.

There are several methods of remembering information and many have adapted interesting ways of organising such. Mnemonics and patterns help humans remember information faster or more efficiently. An example of this is remembering the names of all the planets in our solar system by using mnemonics.

5.4.2 Sensory Overload

Sensory overload is simply too much information i.e. more information than the body can process at the stage of sensory memory as described in the previous sections. This can result from the combination of any of the following senses being overstimulated (Campbell, 2016).

- **Hearing:** Loud noise or sound from multiple sources, such as several people talking at once or a loud fire alarm.

- **Sight:** Bright lights, strobing lights, or environments with lots of movement such as crowds or frequent scene changes on television.

- **Smell and taste:** Strong aromas or spicy foods.

- **Touch:** Tactile sensations such as being touched by another person or the feel of cloth on skin.

Autism or Autism spectrum disorder is a disorder in the brain development which closely relates to the feeling of sensory overload (The National Autistic Society, 2016). It is common for kids to be diagnosed at a very young age (2 and 3 years of age) (AutismSpeaks, 2016). Typically, people who are diagnosed with Autism tend to be extremely sensitive to everyday sensory information which could make them seem to behave inappropriately where in reality they are suffering on the inside (The National Autistic Society, 2016). At this point in time, Autism is one of the fastest-growing developmental disorders in the U.S. and there seems to be no cure or detection and prevention of it (Autism Speaks, 2016).

As the science of sensory marketing is becoming ever more important to the industry, many companies are employing campaigns that strive to stimulate the five senses of consumers in order to capture their attention in more innovative ways (Harvard Business Review, 2016). This however could negatively impact the autism sufferers and also increases the chances of being overloaded information as we go about our daily activities.

5.5 INTELLIGENT PERSONAL ASSISTANTS

Personal assistants (**PA**) are simply speech recognition software applications embedded in mobile devices and computer operating systems such as Microsoft Windows (Hoffman, 2016) and Apple OSX (Apple, 2016) that attempt to recognise the users request and return relevant results.

Siri speech recognition, Google Now, Microsoft Cortana, Amazon's Echo, and Sirius open-ended speech recogniser are all different takes on speech recognition software. Some applications are specifically designed to run on many platforms e.g. Cortana. On the other hand, other applications such as Apple Siri and Google Now are more specific to mobile platforms.

Tasks conducted by personal assistants include the integration of assistance in human lives to help streamline tasks such as reminders, event scheduling, set appointments and even reminding the user where they last parked their car. These assistants typically have access to a database of knowledge in order to respond with relative replies to the query posed by the user (Bosse et al., 2009). Natural Language Processing (NLP) techniques are applied to interpret the query, fetch relevant results and correspond with the user (McShane, 2009).

Typical examples of tasks or services conducted through the use of NLP include schedule management. For example, setting a calendar event or some reminder, searching in the rule-base to reply to queries, sending push-notifications on interesting news such as traffic updates, shop deals and so on (Macy's, 2016). It could also keep track of personal health such as gauging an average heart rate, calorie in-take and any other similar life logging techniques (Memoto, 2016).

It is common for these assistants to behave autonomously and assist with an individuals' workload. These could provide many benefits to the extent which we could see such software build up a profile about the user and adapt to their continued use. For example, Google Now uses Cards as notifications about recent events, current interests, and live. It is worth noting however, that in order to use these type of assistants, one is frequently required to enable accurate position tracking which may introduce cyber-security issues (Kenny, 2016).

There are also other situations in which PAs are suitable, apart from those deployed for personal use. Workplaces have seen potential in applying PAs and multi-agent systems (**MAS**) (Mellon, 2016)to improve the workflow of a company. A multi-agent system is nothing more than a system with multiple agents working towards the same or different goals while existing within the same context. Due to the limitations and problems that arise with groupware and other collaborative tools, PAs and MAS have been proposed to overcome such issues **(Paraiso & Barthès, 2006, p. 1)**.

The ultimate goal of a PA is to reduce the cognitive load for users which would allow them to focus on more important tasks. It is obvious to say that PAs are most applicable for carrying out some routine or trivial task that need not take a chunk of time of our day and thus, could be delegated to the PA to handle. By allowing the PAs to capture information on the fly, their knowledge-management could improve vastly by simply monitoring user behaviour.

Traditional interfaces (graphical-oriented interfaces) are quite inappropriate or does not appeal to many people due to the lack of attention paid to the interaction aspect and the experience one must endure when navigating the interface. Kölzer (1998) had defined a conversational interface as that being similar to the traditional one **(Kölzer, 1998)**, with the exception that the user must communicate verbally with the system to accomplish something **(Paraiso & Barthès, 2006, p. 2)**. These types of interfaces allow users to directly state what they would like in their own words which serves as a more natural way to communicate with an interface. In the end, from an interaction design point of view, these type of interfaces are expected to improve the quality of assistance provided by learning about the user and ultimately reducing the cognitive workload the user must endure.

PAs and MAS are also beneficial for R&D projects since there are countless tasks that could be monitored in the background by such software, allowing the user to focus on their actual workload and not have to waste time on pointless tasks which are time consuming. his will improve project development time, reduce errors since the software would be tested to ensure no failures and also allows for faster collaboration with colleagues by delegating tasks via speech command **(Paraiso & Barthès, 2006, p. 2)**.

5.6 ARTIFICIAL INTELLIGENCE

5.6.1 Fuzzy Logic

There are a variety of applications of fuzzy logic that many have used for experimental and commercial use. Such applications range from consumer products such as cameras, washing machines and air-conditioner systems.

As **Cádiz** explains, Fuzzy logic is represented as multi-valued or many-valued logic. It takes on from Boolean logic which represents On or Off, 1 or 0, black or white **(Cádiz, 2006, p. 69)** etc. It is closest in spirit to human thinking and closely relates to the theory of fuzzy sets, which is a theory relating to the classes of objects with boundaries and memberships relying on a matter of degree.

Human decision-making is approximately the same as fuzzy decision-making since it provides an intuitive method to describe a system in human terms and creates effective models from system requirements. A fuzzy rule is an if-then expression which is made up of the antecedent and the consequence **(Abdul Hamid & Athiar Ramli, 2014, p. 531)**. An antecedent generally refers to the constituents before the comma while the consequence is usually the remaining constituents. An example of such fuzzy rules is described in Example 1.

Example 1. *Fuzzy Rules*

A. *if temperature is **high**, then slightly **reduce** the gas intake;*
B. *if the pavement is **very wet**, then **moderately reduce** your driving speed* **(Valenzuela-Rendón, n.d., p. 346).**

Boolean logic involve values such as 1 or 0, Yes or No, Black or White. Fuzzy values on the other hand are interpreted as values lying within the range of 0 and 1. These values are represented as **'degrees of membership'** since a value may, in the case of men's heights, partially belong to the set of short men and partially to the set of average men. Fuzzy values are represented as a shade of gray that exists between the colours black and white **(Negnevitsky, 2005, p. 88)**.

***Figure 15** - Range of logical values in Boolean and Fuzzy Logic: (a) Boolean logic; (b) multi-valued logic*
(Negnevitsky, 2005, p. 89)

5.6.2 Fuzzy Logic Controllers

5.6.2.1 Defining Hedges

Hedges may be considered as limits or boundaries for some range of values. The word is self-explanatory; to define a hedge refers to defining a range or limit on some value. There are many mathematical formulas which determine a hedge. The table below is specific to certain linguistic variables which define hedges.

These linguistic variables represent a state or value represented within the degree of membership/truth. As previously described, rather than Boolean logic - Fuzzy Logic deals with multi-values lying between 0 and 1 which represent degrees of truth or membership which may be normalised by dividing the maximum value by itself for e.g. consider the computer colours RGB (simply perform division).

a. R – 247/255
b. G – 10/255
c. B – 0/255

We typically use trapezoids or triangles to represent the boundaries for the fuzzy values. Linguistic variables are assigned to specific ranges which are defined by the hedges. This would result in overlapping of values which would show us the result of applying hedges onto values **(Negnevitsky, 2005, p. 97)**.

Hedge	Mathematical expression	Graphical representation
A little	$[\mu_A(x)]^{1.3}$	
Slightly	$[\mu_A(x)]^{1.7}$	
Very	$[\mu_A(x)]^{2}$	
Extremely	$[\mu_A(x)]^{3}$	
Very very	$[\mu_A(x)]^{4}$	
More or less	$\sqrt{\mu_A(x)}$	
Somewhat	$\sqrt{\mu_A(x)}$	
Indeed	$2[\mu_A(x)]^2$ if $0 \leqslant \mu_A \leqslant 0.5$ $1 - 2[1 - \mu_A(x)]^2$ if $0.5 < \mu_A \leqslant 1$	

Figure 16: Table 1.1. Linguistic Variable Hedges defined as Mathematical Properties (Negnevitsky, 2005, p. 97)

5.6.3 **Fuzzy Sets and Operations**

Crisp sets are those sets where an element is either a member of the set or not.
For e.g. **a cat is part of the feline family. Dogs are not.**

A Fuzzy set is a set which isn't crisp but has a clearly defined boundary. It allows elements to be partially in a set **(Valenzuela-Rendón, n.d., pp. 346–347)**. As described in *Figure 3*, two sets of days may be obtained. *Weekdays* and *Weekend Days*; however Friday is partially a W*eekend Day* and partially a *Weekday* (Simulink M. &., Foundations of Fuzzy Logic, 2015).

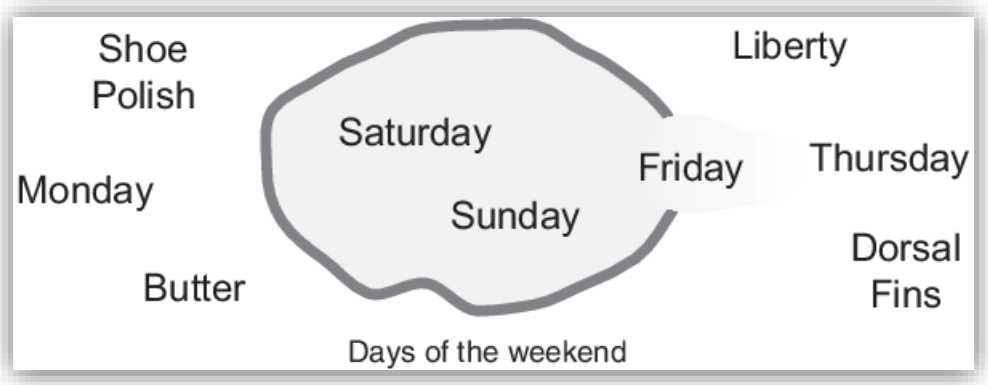


Figure 17: Fuzzy Sets: A set of days comprising a weekend (Simulink M. &., Foundations of Fuzzy Logic, 2015).

The element value can range from 0 to 1 (where 0 is not an element of the set and 1 is a member of the set in Boolean logic). We represent the membership function as a relationship between values of the element and its degree of membership in a set **(Negnevitsky, 2005, p. 89)**.

Fuzzy operations include:

a. *Complement*
b. *Containment*
c. *Intersection*
d. *Union*
e. *And so on…*

In general, when we apply fuzzy operations and hedges to obtain fuzzy sets we get in return, linguistic descriptions similar to our language **(Negnevitsky, 2005, p. 103).**

5.6.4 **Fuzzy Rules**

Fuzzy rules are nothing more than a set of if-then rules which, based on the antecedent, a consequent occurs. For e.g. if x is A then y is B where 'x' and 'y' are linguistic variables and 'A' and 'B' are linguistic values. These variables and values allow us to program a system that tries to follow natural language and attempt to behave similar to a human brain. This is also known as a decision-based system **(A. Shleeg, Aishalaa M. Ellabib, 2013, p. 696)**.

A good example of a rule-base is taken from the paper of **(Kaur & Kaur, 2012, p. 324)** where the rules are based on three membership functions: Temperature, Humidity and Compressor Speed. In this case, the system takes two inputs and generates one output (the compressor speed). An excerpt of the table is shown below; if you split the table vertically and label the first two columns (Temperature, Humidity) as Input and the last column (Compressor Speed) as Output. You should realise that in the case that the temperature is **Very Low** and humidity is **Sticky** then the compressor speed is set to **low** *(see **Rule 4**)*.

Rules	Temperature	Humidity	Compressor speed
1.	Very Low	Dry	Off
2.	Very Low	Comfortable	Off
3.	Very Low	Humid	Off
4.	Very Low	Sticky	Low
5.	Low	Dry	Off

Figure 18: Rule base excerpt from (Kaur & Kaur, 2012, p. 324)

5.6.5 Fuzzy Inference System (FIS)

The Fuzzy Inference System is a process, generally comprising of five stages (Simulink M. &., Fuzzy Inference Process, 2015), **(Sumalatha, Ramani, & Lakshmi, 2011, p. 11)**. This process formulates the mapping from a given input to an output **(Negnevitsky, 2005, p. 106)** by involving all the pieces described in *sections 5.2.1, 5.2.2 and 5.2.3*.

5.6.5.1 Fuzzy Inference Process

Two types of fuzzy models exist for the fuzzy inference process. The Mamdani-type and Sugeno-type **(Abdul Hamid & Athiar Ramli, 2014, p. 533)**. Each model's inference process is almost identical except for the defuzzification stage.

Mamdani-type applies the centroid calculation method in order to obtain the centre of gravity while the Sugeno-type uses the weighted average of the data method **(Abdul Hamid & Athiar Ramli, 2014, p. 536)**.

5.6.5.1.1 Fuzzification of all input

The first step is to take the inputs and obtain the degree of membership to which each element belongs to the appropriate fuzzy sets with membership functions **(Sumalatha et al., 2011, p. 2)**.

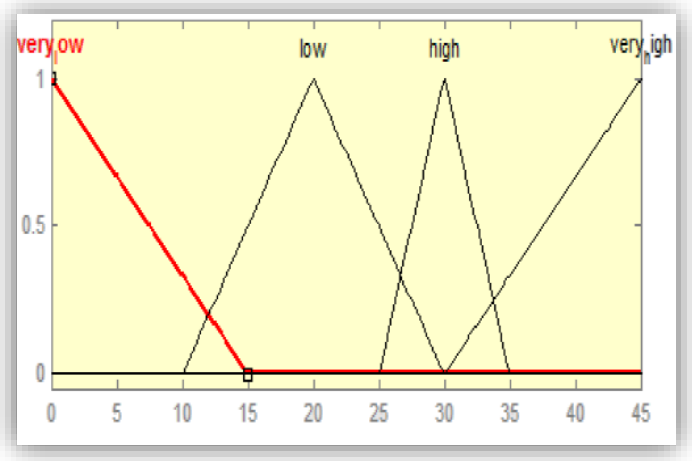


Figure 19: *Membership function for temperature* (Kaur & Kaur, 2012)

5.6.5.1.2 Application of Fuzzy Operators in the antecedent

This stage takes the fuzzified inputs and applies them to the antecedents of the fuzzy rules. If the antecedent of a given rule consist of more than one part, we apply the fuzzy operator such that we obtain one number to represent the result **(Negnevitsky, 2005, p. 109)**.

Fuzzy Logic Toolbox supports two-built in AND methods: *min (minimum)* and *prod (product)*. Also supports two-built in OR methods: *max (maximum)* and *probabilistic OR (probor), also known as the algebraic sum* (Simulink M. &., Fuzzy Inference Process, 2015). s

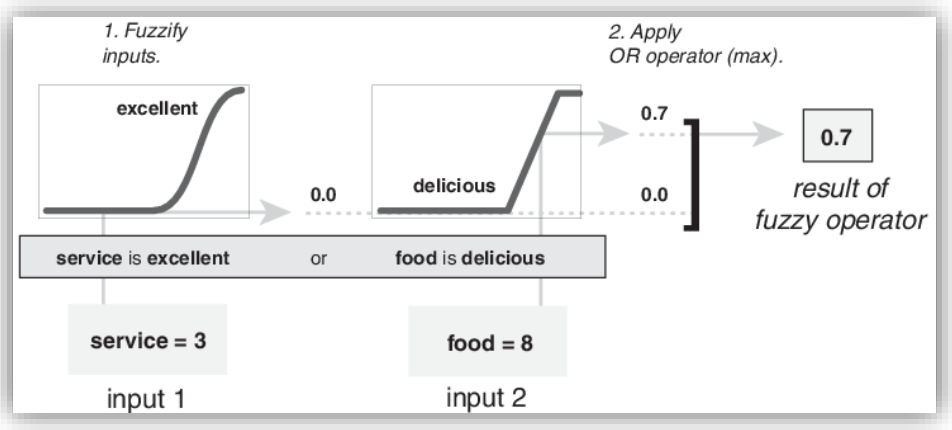


Figure 20: *Application of Fuzzy operators* ***(Simulink M. &., Fuzzy Inference Process, 2015)***

5.6.5.1.3 Implication from the antecedent to the consequent

Every rule has a weight which is a value between 0 and 1. After each rule has been assigned a proper weight, the implication method is implemented.

The consequent is yet again another membership function which also has a weight assigned that describes the characteristics regarding it (Simulink M. &., Fuzzy Inference Process, 2015)**, (Sumalatha et al., 2011, p. 11)**.

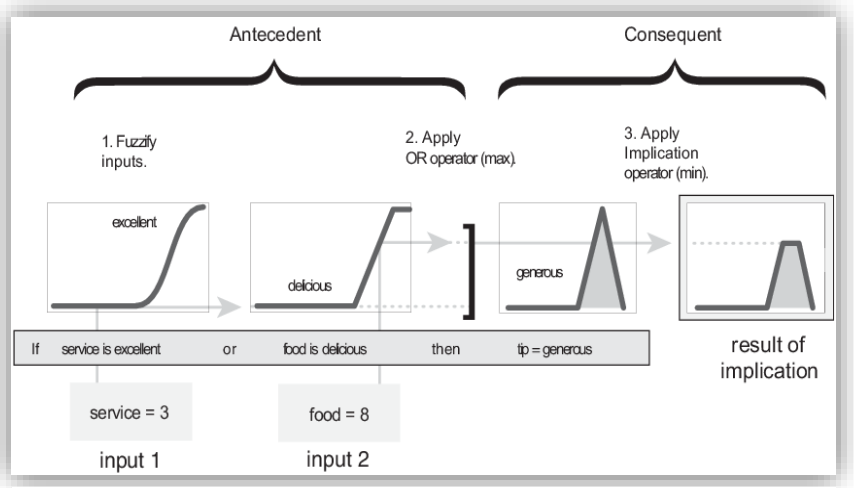


Figure 21: *Applying the Implication Method* ***(Simulink M. &., Fuzzy Inference Process, 2015)***

5.6.5.1.4 Aggregation of consequents across the rules

We must combine rules in order to make decisions. The aggregation process combines all output fuzzy sets into one single fuzzy set **(Negnevitsky, 2005, p. 110)**. Three built-in methods are supported:

a. max (maximum)
b. probor (probabilistic OR)
c. sum (sum of each rule's output set) (Simulink M. &., Fuzzy Inference Process, 2015)

Aggregation only occurs once for each consequent, just before the last stage – defuzzification. Unless the aggregation method is commutative, then the order of execution is unimportant (Kumar & Senthil, 2015).

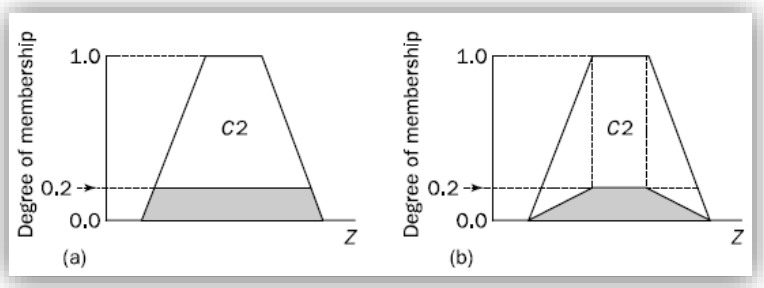


Figure 22: Clipped (a) and scaled (b) membership functions (Negnevitsky, 2005, p. 110)

5.6.5.1.5 Defuzzification

It is desired to come up with a single crisp output from a FIS. For example, when someone is trying to classify how much to tip, the FIS must return a crisp value in order to find out the correct tip. Nevertheless it is obtained in the latter stage, defuzzification (Knapp, 2015).
Two techniques may be applied in this stage which involve both Mamdani-Style and Sugeno-Type Inference (Knapp, 2015)**, (Zaher, Kandil, & Fahmy, 2014, p. 3015), (Salman AbdulWahed & Seno Ismat, 2012, p. 297)**.

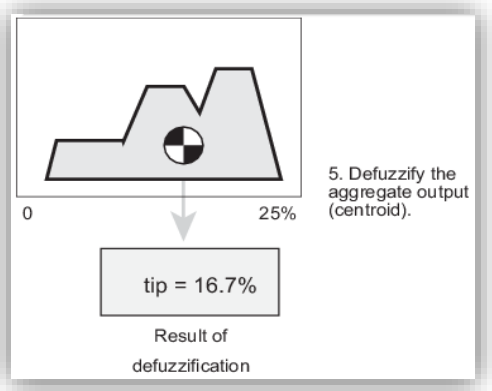


Figure 9 – Result of defuzzification (Simulink M. &., Fuzzy Inference Process, 2015)

5.6.5.2 Mamdani and Sugeno-Type FIS

Both approaches have specific roles in certain applications however in general cases, Sugeno-type proves to perform better in a variety of situations such as dynamic non-linear systems however this can only be used in (MISO) *multiple input single output systems* **(Salman AbdulWahed & Seno Ismat, 2012, p. 297)**.

The Mamdani-Style however allows the use of both MIMO (*multiple input multiple output*) and MISO systems. It also allows for intuitive formalisation and interpretability of fuzzy rules in a more human-like manner **(Negnevitsky, 2005, p. 114), (Kaur & Kaur, 2012, p. 323)**.

The most fundamental difference between both FIS is the manner in which the crisp output is obtained. The Mamdani-Style maintains the defuzzification technique using membership functions whereas Sugeno-type FIS uses weighted average to compute the crisp output **(Kaur & Kaur, 2012, p. 323),** (Knapp, 2015) (Simulink M., 2015)**,** (MATLAB, 2015)**.**

5.6.5.2.1 Mamdani-Type FIS

Mamdani FIS is more widely used, mostly related to the fact that it provides reasonable results with simple methods and its intuitive nature in terms of rules **(Zaher et al., 2014, p. 3015), (Jassbi, Serra, Ribeiro, & Donati, 2015).**

The Mamdani approach uses the center of mass (gravity) which slices the aggregated set into two equal masses to come up with one crisp number **(Negnevitsky, 2005, p. 111)**. The output of Mamdani FIS has output membership functions.

In turn, **Mamdani Method** is more computationally intensive since it is less flexible in system design in comparison to Sugeno FIS since the latter can be integrated with ANFIS tool to optimise the outputs **(Kaur & Kaur, 2012, p. 323)**.

$$G = \frac{\sum A_i g_i}{\sum A_i},$$

Figure 23: *Total gravity center G is calculated as above (Valenzuela-Rendón, n.d., p. 350)*

5.6.5.2.2 Sugeno-Type FIS

The process is very similar to the Mamdani-Type FIS however the only difference is the way in which the crisp output is generated from the fuzzy inputs. The Sugeno-Type FIS uses the weighted average to compute the crisp output which serves as a great advantage for applications where it is a non-linear dynamic system **(Zaher et al., 2014, p. 3015), (Negnevitsky, 2005, p. 114).**

It is worth noting that this FIS can also integrate with ANFIS tool to optimise the outputs for individual users or environments **(A. Shleeg, Aishalaa M. Ellabib, 2013, p. 323)**.

In general **Sugeno Method** is considered more computationally effective, flexible and more efficient than a Mamdani FIS **(Salman AbdulWahed & Seno Ismat, 2012, p. 297)**.

$$WA = \frac{\mu(k1) \times k1 + \mu(k2) \times k2 + \mu(k3) \times k3}{\mu(k1) + \mu(k2) + \mu(k3)} = \frac{0.1 \times 20 + 0.2 \times 50 + 0.5 \times 80}{0.1 + 0.2 + 0.5} = 65$$

Figure 24: *wA (Weighted Average) (Negnevitsky, 2005, p. 114)*

5.6.5.2.3 Comparing both methods through review of papers

Although Mamdani-Type and Sugeno-Type FIS produce similar if not the same output, various authors have used both methods to compare or else simply chosen one of them.

A good example is that of **(Cádiz, 2006, p. 71)** where his choice of FIS is the Mamdani-Type, the triangular membership function that is produced is very similar to those produced in and below.

(Zaher et al., 2014, p. 3015) sums up in the motivations for their paper in the application of Mamdani FIS and Sugeno FIS that from the three key points that determine Sugeno FIS is more flexible, computationally effective and more adequate for functional analysis. It concludes that Sugeno FIS is always more efficient than a Mamdani FIS.

This argument is supported by various authors **(Valenzuela-Rendón, n.d.), (Kaur & Kaur, 2012), (Kamboj & Kaur, 2013), (A. Shleeg, Aishalaa M. Ellabib, 2013)** whereby they prove by comparison, both Mamdani and Sugeno-Type FIS implementations for solving their problem. In all papers, Sugeno proves to perform most efficient in terms of results and max capacity of the system. This however does not discourage the use of Mamdani-Style inference as it has its own advantages such as the flexibility of using both MISO and MIMO systems **(Salman AbdulWahed & Seno Ismat, 2012, p. 297)**.

6 METHODS

6.1 INTRODUCTION

This section discusses the research and methodological approach for the system developed in order to meet the aims and objectives. The first section discusses the research strategy while the rest explores the implementation of the system. The latter section goes into technical detail about how the input was obtained and mapped using Fuzzy logic as well as the design of the visualisation in both desktop display and a fully immersive virtual reality environment. Also included within this section are the steps taken to test the prototypes, develop the questionnaire and carry out the experiment.

6.2 RESEARCH STRATEGY

Research started out using the Internet. Some more information was gathered from preliminary research and a book called Consumer Behaviour (Kanuk, 2016) loaned from St. Martin's Institute of Higher Education Library. After the main idea was finalised, most of the time researching was spent using the following resources:

- Google Books, Google and Google Scholar
- OpenAthens Library
- ResearchGate

Search Engines

With respect to OpenAthens, I found both Google Search and Google Scholar to be more suitable in terms of true positive results obtained when searching for published papers, books and journal matching my scope of research.

I initially started researching Fuzzy Logic Systems as suggested by my supervisor. Topics covered included several applications with no general restriction due to the wide applicable use of these type of systems. At some point, the term Fuzzy Logic results were near-saturated, thus required me to vary my search terms. It didn't take me long to encounter keywords that returned copious amounts of research papers and published articles with the use of Fuzzy Logic Systems.

Having found a wealth of information to discuss the topic, researched progressed onto Psychology, Consumer behaviour and Virtual Reality. The terms measuring human behaviour were used as a combination with other terms such as computing, art, stress impact etc., which led to a lot of research papers that discussed Audio-visual media and similar fields. At this point, other terms such as galvanic skin response and heart rate were included in many queries so as to include human variables in the scene.

Virtual Reality alongside the other terms were easily researched and did not require much effort to conclude sections in the literature review. The final sections researched involved

Intelligent User Interfaces and information in the context of computing and its relation to Sensory Overload.

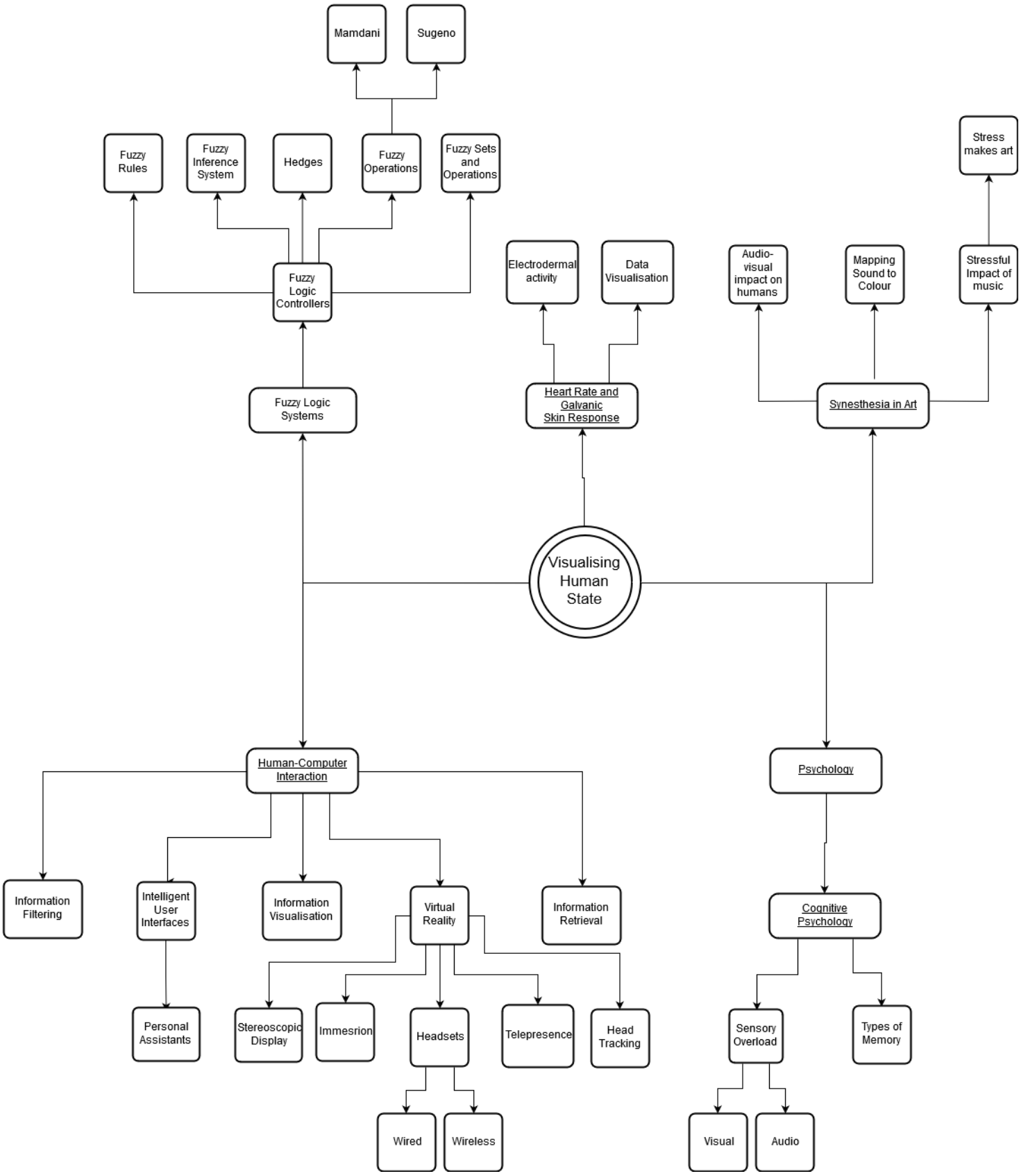


Figure 25: Overview of terms researched

6.3 SYSTEM MODULES

The following sections detail the sub-modules that were used to receive, process and output the visualisation. Said visualization combined the physiological data generated by participants and the music presented to them.

In order to simplify the development process, the system was divided into three modules. These were subsequently further sub-divided into smaller modules/components, then finally combined once each prototype was completed (Divide and Conquer approach). In summary, the system was designed using a top-down approach, starting from the input – going all the way down to the output.

The three primary modules are the:

- Input Module
- Data Processing Module
- Output Module

The **Input Modules**' sole purpose were to read in physiological data via Heart Rate (**HR**) and Galvanic Skin Response (**GSR**) sensors that were connected to the Arduino Uno Microcontroller. This in turn, communicated with the computer via serial communication (USB).

The **Data Processing Module** entailed the processing of human and musical data. This was performed by decoding incoming serial data sent from the Arduino to the computer and subsequently extracting the spectral peaks of the input sound clip at the specified time intervals. The data sent from arduino are considered as samples and were combined with extrapolated musical features every three seconds. The samples and peaks were combined into a list of 32 rows of type string with the HR and GSR samples appended to the end of the string. This was done as the spectral peaks returned from the Arduino consisted of a double array with 32 elements. Once this process is complete, the packets were traversed and fuzzified by passing the data obtained through the Fuzzy Logic Controller. This was then encoded into JSON format and uploaded as a JSON file to the remote server using the FTPClient class (**see 4.3.2. Data Processing Module - JSON Encoder and FTP Connector**).

Finally, the **Output Module** fetched the JSON files from the remote server, decoded the files and stored the data, then mapped the fuzzified data to generate a visualisation. Said visualization was updated in real-time based on the data that had been uploaded to the remote server. The two kinds of displays used were:

- Google Cardboard (Head Mounted Display - HMD)
- Traditional Screen (Traditional Display Environment - TDE)

Both forms of display presented the same visualization, with the notable exception of their build types and some device specific configurations. The main idea was that the Output Module processed obtained data from the music and the participants' physiology (human input), and then displayed a visualisation corresponding to the four main properties:

- Colour
- Size
- Pulse Rate
- Speed

The first two properties (colour and size) were mapped directly to the Frequency and Amplitude extracted from the music being heard (**see Figure 3: Data Processing Module**) by the participant. Similarly, the Pulse rate and Speed were also directly mapped from the HR and GSR samples respectively.

Further, Figure 1 depicts the system overview showing how data was sent from the Input Module all the way down to the Output Module which was then repeated till the experiment terminated (acting as a feedback loop (**see 4.3 – Methods: Experiments**). It is worth noting that each module is developed using different programming languages, due to the various requirements set in place by the hardware vendors (**see Appendix: Section 3 – Methods: Hardware Configurations**).

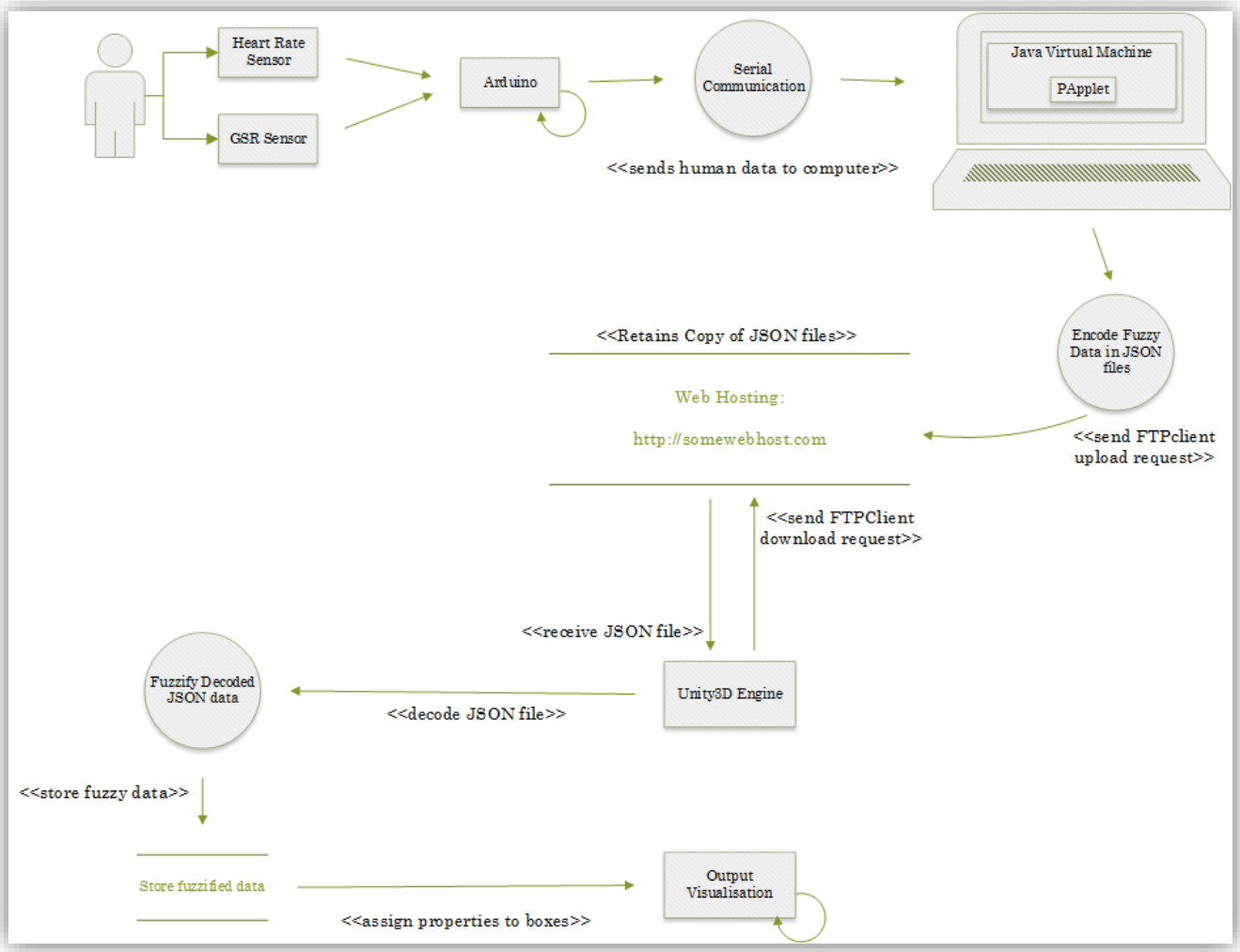


Figure 26: High-level System Overview – all three modules

The reader is advised to look at **Appendix: Section 1 – Methods** for all the code used to make this system function. It is important to understand that the combination of platforms do not work without the Arduino and sensors, thus I have provided some snapshots and videos to get an idea of the experiments that took place.

6.3.1 Input Module:

The input module required the participant to wear sensors that read their data in real-time and sent it to the Arduino for further processing. The human input was gathered with the use of the two sensors: the Galvanic Skin Response (**GSR**) which measured skin conductance levels and a Heart Rate Sensor (**HR**) which kept track of the participants' heartbeat, measured in beats per minute (**BPM**).

Data began being transmitted to the Arduino controller once the participant had worn the sensors and a serial connection had been established via USB. In order to interface with Arduino, one must download the software IDE (Arduino, Download the Arduino Software, 2016) in order to receive data from the controller. Once the IDE is setup and all the necessary code has been compiled and uploaded to the Arduino (given that the Arduino is also set up appropriately), human input will be able to flow from the sensors to the Arduino, straight into the laptop.

Both sensors required different voltages to operate and communicate with the Arduino through different channels. That is the GSR sensor communicated via the Analog input bank whilst the HR sensor communicated via the Digital input bank. This in turn, allowed the human input to be read without signal interruption or additional noise when monitoring the human input (**see Figure 4: Circuit Diagram for Arduino, Breadboard and Sensors**).

Thus, both sensors were connected through the same Arduino which simultaneously read data from the sensors by powering the GSR sensor with 3.3v supply, its own ground and a direct connection with the A3 (Analog Input 3) bank on the Arduino board. This can be seen in the circuit diagram in Figure 3. Conversely, the HR sensor was assigned a different ground bank by passing the load through the breadboard while attaching the input cable to the #2 (Digital Input 2) slot (dc42, 2016).

It is worth emphasising the distinction indicated above. That is, the GSR sensor used Analog input whilst the HR sensor used Digital input when communicating with the Arduino. The code used to read in human input can be seen in the **Appendix: Section 3 – Methods: Arduino Program Code**. It is worth pointing out that the Arduino was programmed to send data with a header or just 0 since the HR sensor sends two values

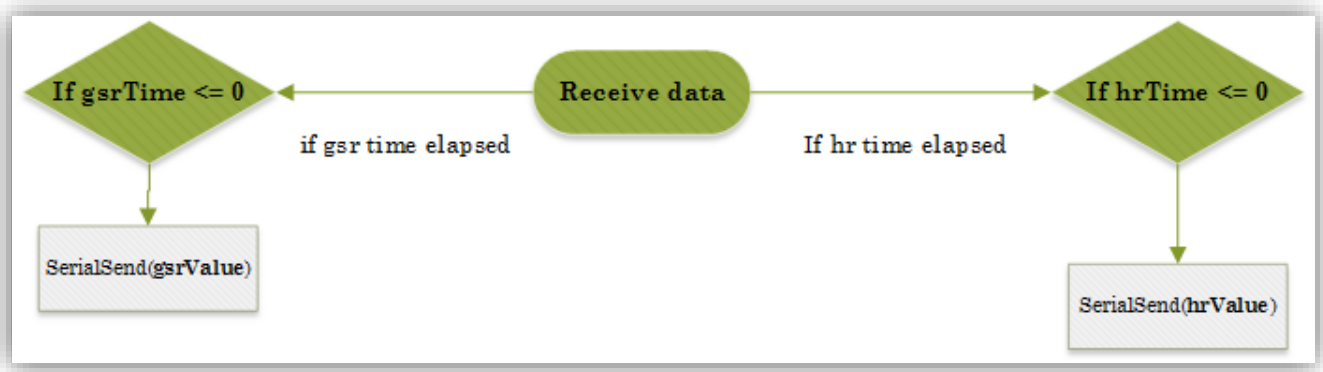


Figure 27: Decision tree for the classification of data sent to communication port

(1023 and 0). Thus, the header was used to indicate what type of data has been received, be it GSR, HR, some digit or a combination of the aforementioned (see **Figure 18**). This concept allowed me to parse the serial data more efficiently in the Data Processing Module, which is described in greater detail in the next section.

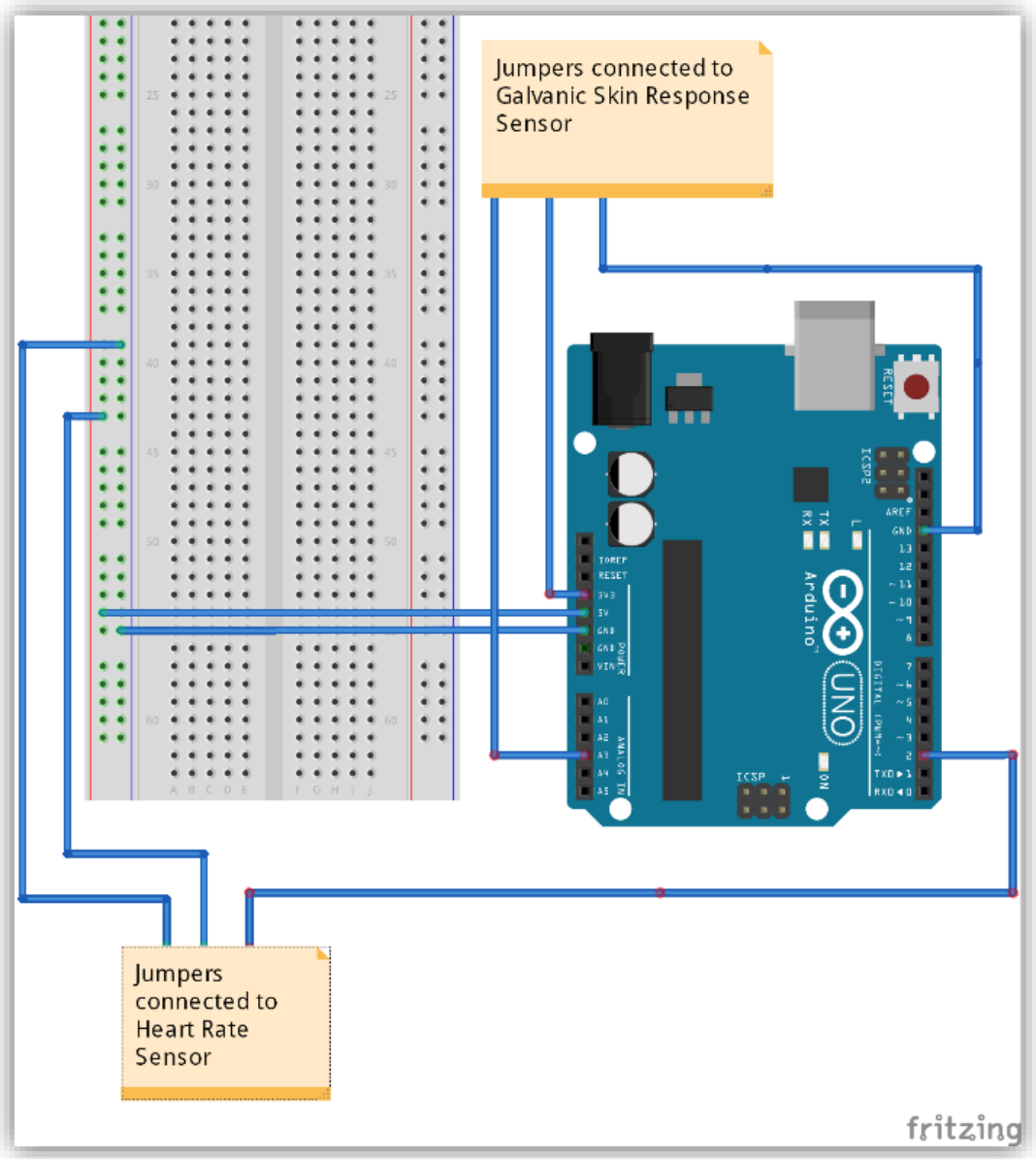


Figure 28: Circuit diagram for the Arduino and the method of setting up the GSR and HR sensor for sending data simultaneously (Fritzing, 2016)

6.3.2 **Data Processing Module:**

The Data Processing Module acted upon the data obtained in the Input module and further processed it using features such as an Arduino Data Interpreter Algorithm, Fuzzy Logic Controller (FLC), JSON encoder and the FTP connector. These were all built using Java except for the FLC which uses its own plugin to design the model in the Eclipse IDE.

Each of these features processed a piece of data and eventually stored it for further processing. This was repeated since each process acted as a separate event, with each event chained to the next, occurring every three seconds with the use of a ScheduledExecutorService in Java (Bender, 2015)**)** until the experiment terminated. The reader is advised to refer to Figure 4: Data Processing Module flow which represents how the musical properties and human input are fuzzified and wrapped into a JSON file which was then uploaded to the server (using a web host for remote storage).

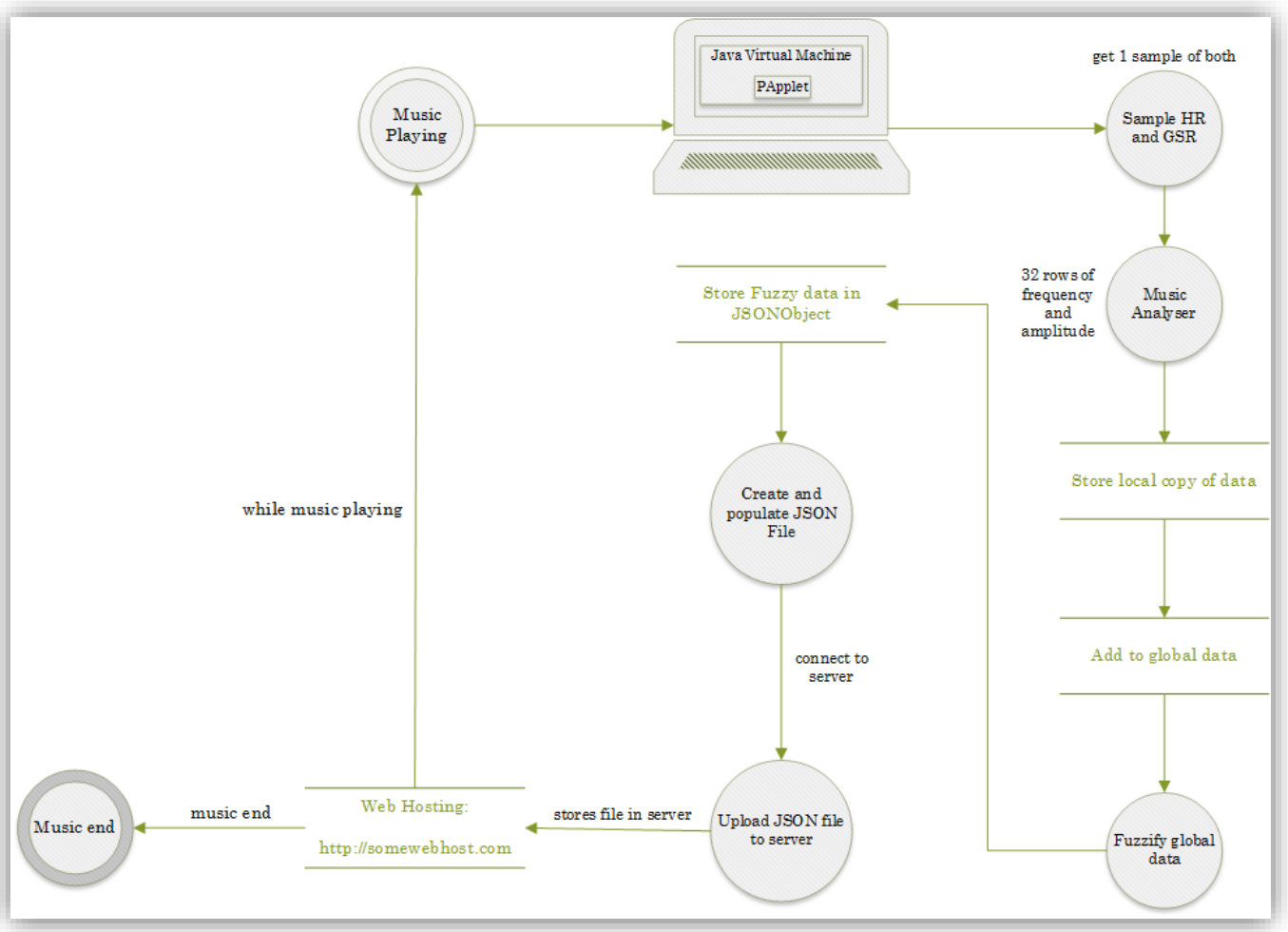


Figure 29: *Process begins at Music Playing which starts in Eclipse (JVM). Data flowing from input module is sampled in this module and is passed onto the analyser which further stores the data to be fuzzified, wrapped in JSONObject, stored in a json file and uploaded to the server to be retrieved in the Output module.*

6.3.2.1 Arduino Data Interpreter Algorithm

The SerialPortEventListener interface (Oracle, 2016) provided an unimplemented method - serialEvent() which enabled the data received to be decoded through the communication port that the Arduino was connected to. An inner class called PortReader() (**see Figure 21: Class diagrams representation of the Data Processing Module sub-system**) which implemented the SerialPortEventListener interface, was used to interpret the data received when interfacing with the Arduino.

The serialEvent() method provided direct communication with the port specified in the instantiation of the ComPort(), and was used to classify the last received signal into either a GSR or HR list of samples. The reason for classifying said data was due to the readString() method which converted the byte data into a string. When the data was sent to the communication port, it was segmented into pieces by default and then converted back into legible format.

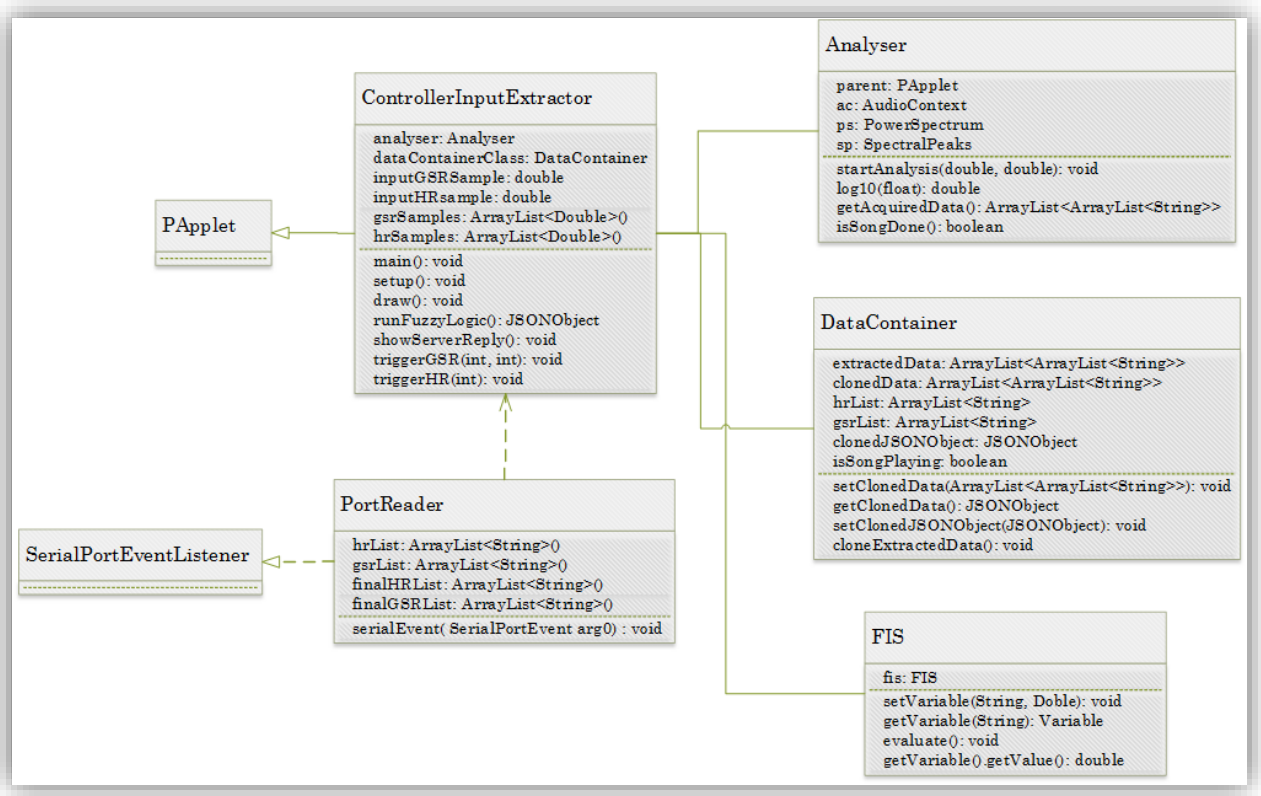


Figure 30: *Class diagrams representation of the Data Processing Module sub-system*

An issue encountered was that upon printing the received data, the SerialPortEventListener seemed to append a new line every few characters. To alleviate this issue, an algorithm was implemented that kept track of the last received data packet and then checked whether the next received piece of data contained a header or just a digit (**see Figure 22: Classification process for assigning the last received packet from the Arduino**).

Thus, if the data started with the letter 'g' then it denoted a GSR signal, which was then stored in a list and safe to use for further processing. On the other hand, should the data have started with the letter 'h' then it would have been classified as a HR signal and stored it in its own container, similar to the GSR signal. Lastly, if the piece of data was a digit, then that particular packet would be appended to the previous packet received as it must have been split up into two packets (see **Appendix: Section 1 – Methodology: Data Processing Module – Classifying Serial Data**).

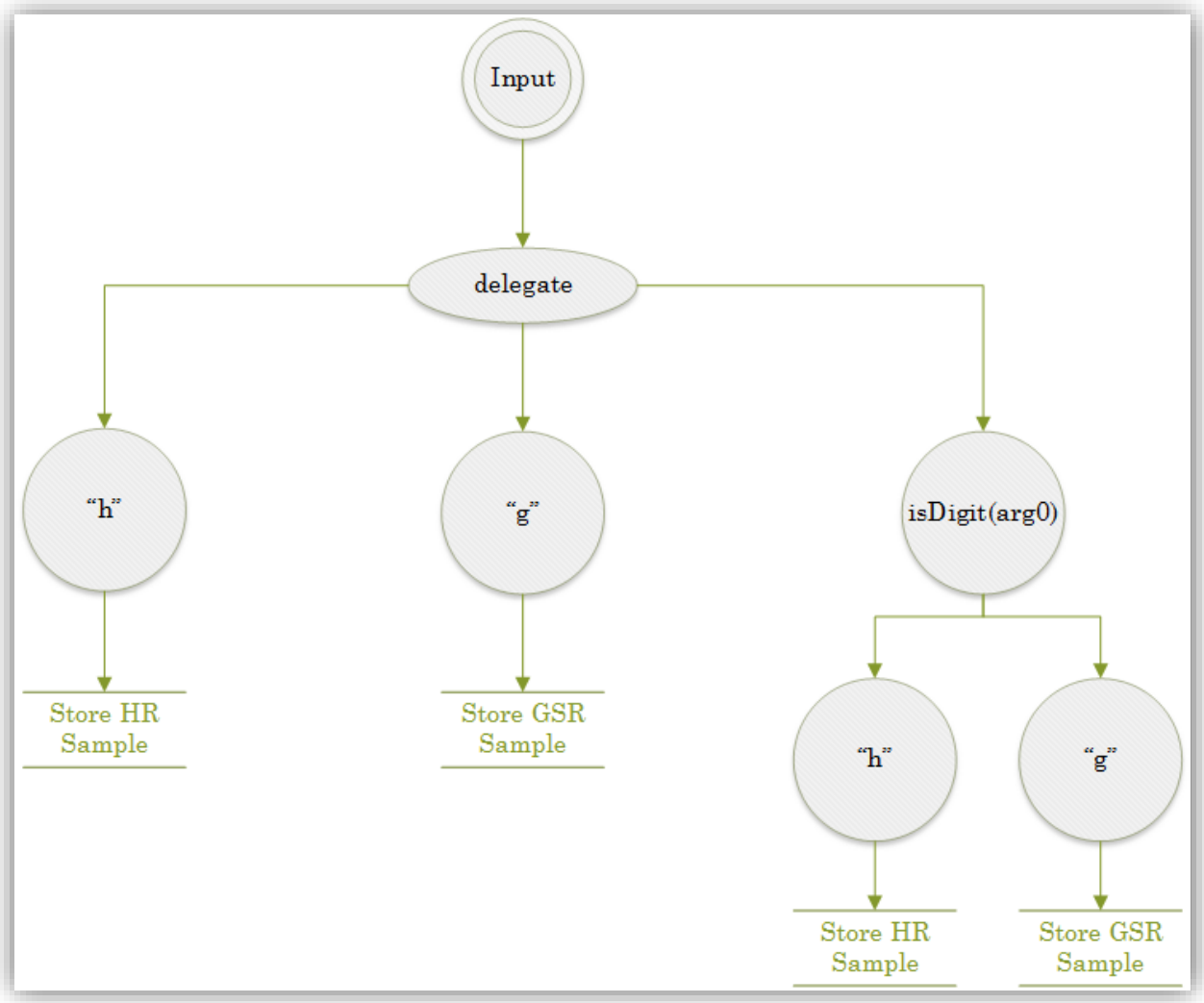


Figure 31: *Classification process for assigning the last received packet from the Arduino*

6.3.2.2 Fuzzy Logic Controller

The Fuzzy Logic Controller featured in many parts of this project and is considered to be one of the main fields that motivated this research. As explained in the Literature Review section a good application for Fuzzy Logic is mapping one domain to another. As previously explained in the Preliminary Project Report, the library chosen for this part of the system was jFuzzyLogic, which is an open-source Fuzzy Logic Controller built in Java and was developed by Pablo Cingliani, and Jesus Alcala-Fdez **(Cingolani & Alcalá-Fdez, 2012), (Cingolani & Alcalá-Fdez, 2013)**. It complies with part 7 of the IEC 61131 norm and provides an easy-to-use system which only requires the programming of Inputs, Outputs, Rule Blocks and Conditions. This software also comes with an Eclipse plugin that allows users to work with more-legible syntax when implementing a Fuzzy Logic System.

For more information about how Fuzzy Logic works, the reader is advised to visit Section 2: Literature Review and read through the technical details about the general concept, descriptions of its components and a couple of model comparisons. For the rest of this sub-section, the implementation details of the fuzzy logic controller with respect to its role in the system will be discussed.

As explained in the introduction above, Fuzzy Logic can be used to map one domain to another. It is widely understood that mapping the Aural to the Visual domain results in interesting creative artefacts. This project took this notion one step further and added human input to the equation. This is further described in Section 1: Deliverables and Objectives.

By referring to the System Module section, the visualisation was defined by four properties – colour, size, pulse rate and speed. Thus, these values acted as the output properties for the Fuzzy Logic Controller (**see Figure 23: Fuzzified Output Variables**). However in order to have output, there must exist some input. Therefore the input would be (also previously described in the System Module section) Frequency, Amplitude, Heart Rate and Galvanic Skin Response.

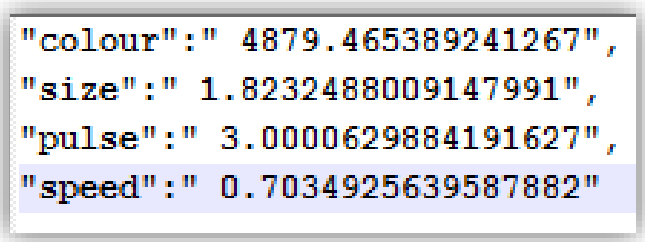


Figure 32: *Sampled fuzzified Output Variables*

Rule Blocks within the Fuzzy Logic System contained the calculation method and rules to be evaluated at runtime. The evaluation process considered all the factors described in Section 2: Literature Review – Fuzzy Logic Controller which evaluated all the values in order to map from *values* to a *visual* context (**see Appendix: Section 3 – Methods: Mapping the physical to the virtual world**).

The rule blocks defined which method was used to calculate the weights of the applicable rules, then searched through the rule base to activate the appropriate regions with respect to the input. As indicated in Figure 9 and 10, the input values range from some minimum to some other maximum and likewise, the same applies for the output values. The Fuzzy process began when all the inputs and outputs had been appropriately set and rules with non-overlapping terms had been defined. The Fuzzy Inference System (or **FIS** for short) then evaluated and activated regions to deduce a final crisp value (output) for each input.

It is worth noting that for the purposes and requirements of the system, a Mamdani-type method was used as it did not seem to suffer in terms of bottlenecks. Further, considering that the Sugeno-type method only produces one output value – it did not seem to be applicable in this scenario. Apart from this, I faced some issues relating to thread handling (**see Section 8: Limitations**). An example of a resulting weight may be seen in Figure 11: Activated regions displaying the final weighting output.

6.3.2.3 JSON Encoder

JSON Objects are widely used for transmitting data over the internet or through some stream which allows users to encapsulate data into JSON objects, providing them with a simple method of manipulating and storing lists of data in a convenient file format.
A JSON object starts with an open curly bracket '{', and ends with a closed curly bracket '}'. Each JSON object may be nested within itself and therefore the format allows for several nested of objects. However this is ultimately up to the designer to choose how to go about it (**see Figure 9: Example of typical JSON file**).


```
{
   " 11":{
      "colour":" 4922.5608019944975",
      "size":" 1.9415998809725894",
      "pulse":" 3.0000629884191627",
      "speed":" 0.7034925639587882"
   },
   " 12":{
      "colour":" 4737.399718561236",
      "size":" 1.973760857777537",
      "pulse":" 3.0000629884191627",
      "speed":" 0.7034925639587882"
   },
   " 13":{
      "colour":" 4742.529773674851",
      "size":" 1.9918277065927812",
      "pulse":" 3.0000629884191627",
      "speed":" 0.7034925639587882"
   },
   " 14":{
      "colour":" 4737.77026269722",
      "size":" 2.0005065264364363",
      "pulse":" 3.0000629884191627",
      "speed":" 0.7034925639587882"
   },
   " 15":{
```


Figure 33: *Example of a typical JSON file*

Once the JSON parser completed its operations, the fuzzified extracted frequency, amplitude, HR and GSR samples are stored in a JSON object in order to be uploaded to a remote server as a .json file. Please refer to **Appendix: Section 1 – Methods: JSON Encoder** for the JSON encoder code.

As previously described in the System Modules section, each file consisted of 32 rows of type string. Each row consisted of the four fuzzified properties assigned to each box displayed in the visualisation (explained in the Output Module sub-section). This made it convenient for the decoding stage in the Output module since each file is of size 32, enabled me to maintain a set of 32 boxes in the visualisation at run-time. Considering the larger screen on standalone, I had increased the number of boxes active in one screen for the standalone version).

6.3.2.4 FTP Connector

The FTPClient class provided an interface that connected to a remote server and uploaded the files via the File Transfer Protocol (FTP on Port 21) using the FTPClient constructor provided in the Apache Commons Net 3.4 API (Apache, 2016).

This class allowed for a secure connection to be established by logging into the FTP client web hosting service (000webhost, 2016) that hosted the aforementioned JSON files. This was accessed frequently by the unity application when updating the visualisation at run-time (**see Output Module sub-section**).

The web host served as a remote storage location for the JSON files produced during the Data Processing Module. It also allowed the application on the mobile device to update in real-time through the internet (intentionally designed for the Virtual R display).

After successfully connecting to the FTPClient, the process began by creating JSON files which were then stored on the remote server using the FTPClient in Java (using the storeFile() method provided in the Apache Commons API). The reader is advised to refer to **Table 1: The FTP authentication and configuration process** to better understand the process flow of JSON file storage on the remote server.

It is worth noting that due to the ScheduledThreadExecution occurring every 3 seconds, the FTP connection disconnected and reconnected to and from the FTPClient at every upload attempt (**see Appendix: Section 3 – Methodology: Processing Module**). Thus, the process was repeated every 3 seconds for every new file to be stored on the remote server. The files were stored by index, with a following underscore and *'fuzzyData'* appended to the end of the file name (**see Figure 25: File names of Fuzzified JSON Objects**) in order to decode the JSON file more efficiently in Unity3D (see Output Module further on).

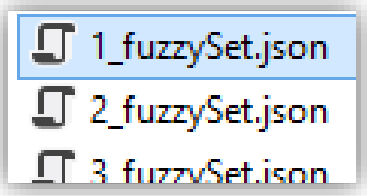


Figure 34: *File names of Fuzzified JSON Objects*

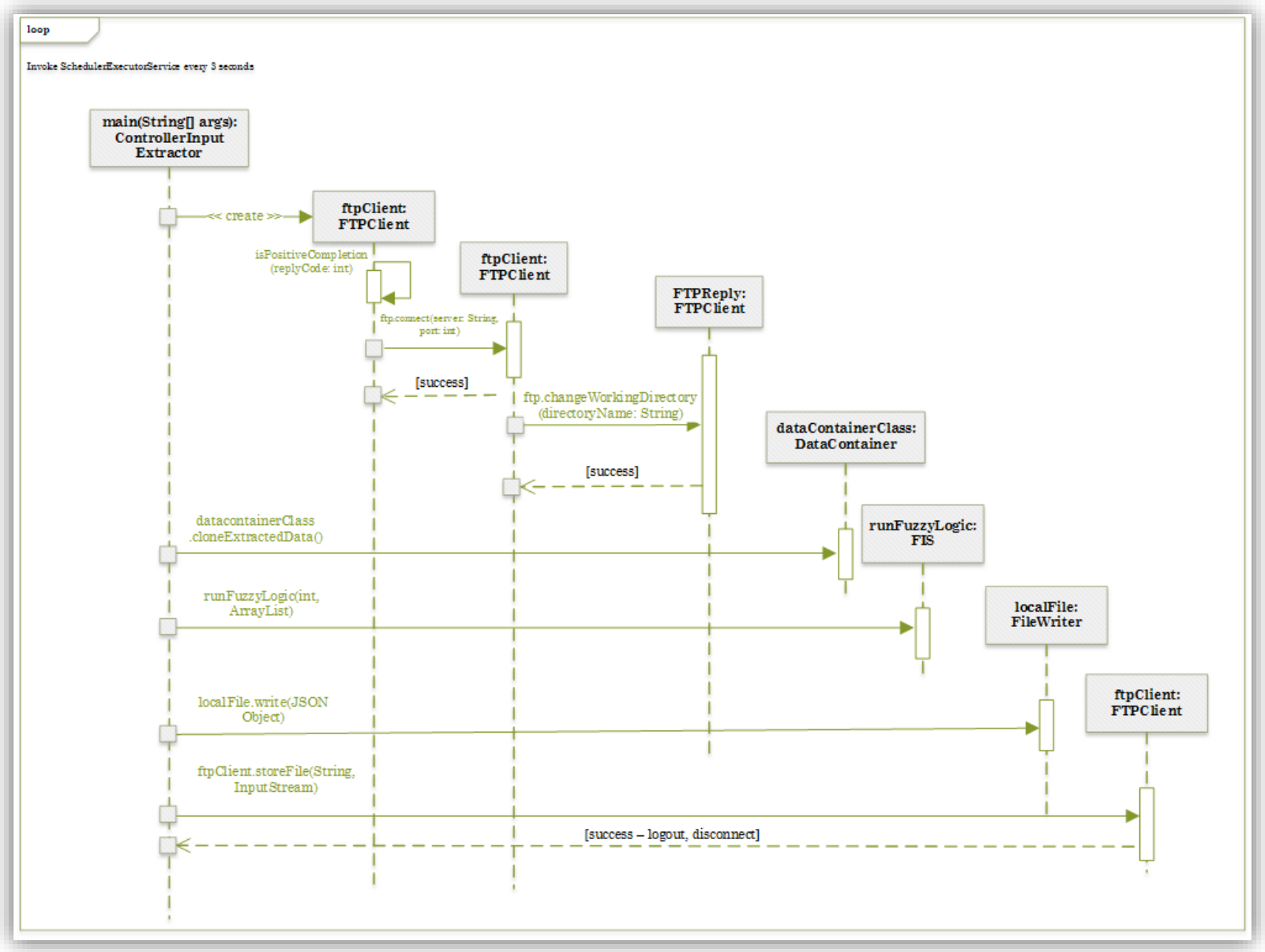


Figure 35: *The FTP authentication and configuration process*

6.3.3 **Output Module**

This section covers the front-end of the system i.e. the output module for the visualisation which was designed using Unity3D in C#. The Output Module's main objective was to map data stored on the remote server onto the visualisation.

- In order to make a comparison between virtual environments and the traditional screen, this module required two builds: one for Google Cardboard and the other for traditional screens, both implementing the same four components: Visualisation of boxes moving to the viewer
- Fetching file from URL
- Decode JSON and store data
- Mapping data to boxes in Visualisation

The visualisation was also interactive by providing the user the ability to destroy the boxes on screen using the trigger button on the left side of the headset. This was done so as to maintain the engagement of the participants. The trigger uses a magnet to shift the magnetic field of the compass used in mobile devices (gyroscope), tricking the machine into thinking that a button event occurred.

6.3.3.1 Computers and Devices used for display

The difference between Google Cardboard and the traditional screen is in terms of the build type and on what the viewers see and the computer form factor. In the case of Google Cardboard, Android build was required to display the visualisation in a virtual environment.

On the other hand, the traditional screen (standalone build) was simply the visualisation in 2D format (**see Appendix: Section 1 – Methods: Computer and Devices used for Output Module**). I removed any prefab (object instance) and any assets (directories/libraries) related to Google Cardboard and replaced the trigger button with a mouse click trigger event which is easily accessible through Unity.

Both the Android and the Standalone builds implemented the same algorithms to run the visualisation, with the only exception being the view. The android build implemented stereoscopic vision, whereby two screens are provided and display the same scene for each screen from a slightly different angle (**see Section 5: Literature Review – 5.3. Virtual Reality**). Whereas the standalone build dropped any Cardboard assets and prefabs and simply showed the visualisation in 2D format on a traditional screen (e.g. laptop screen – **see snapshots from experiments**).

6.3.3.2 Visualisation of boxes moving towards viewer

In order to retain efficiency whilst still making the application interesting, a sense of closure was expected. This means that it was imperative to destroy the boxes created over time since besides becoming computationally expensive, it also does not make sense to retain an instance of the object if it is not in the line of sight of the viewer.

The participant's perspective was such that it emulated being placed in the centre of a sphere, with a large wall directly behind the participant's starting perspective. In practice, this translated into a 180° area wherein the participant could have looked

around and seen the boxes approaching however this did not apply for traditional display environments. In this case participants had approximately 60° field of view, only able to destroy boxes within their peripheral vision.

In order to find out the field of view visible to the participant in Unity, one can simply invoke the *Camera.fieldOfView* which returns the field of view visible to the user in degrees (Unity, 2016).

As described in the previous sections, the boxes were defined by four main properties. Boxes were instantiated and spawned within hemisphere visible to the user at random positions. The boxes gradually approached the participant at constant speed. As the boxes approached they obstructed a greater area of the participant's field of view. To counteract this the participant was able to destroy the boxes coming towards them. The first property – Colour represented the frequency value extracted from the spectral peaks of 3 seconds clips of the musical piece. Given that the spectral peaks returned a 2D array, the second parameter, amplitude was used to represent the Size property of the box (mapping loudness to size).

The third property was pulse rate which mapped heart rate to pulse rate (i.e. that it scales up and down the size of a box repeatedly at the specified rate). **See Section 7.5: Methods – Third Party Libraries** for the tool used to do so. Finally, the fourth property was the speed which was a mapping from the GSR obtained from the participant that affected the visualisation in two ways. The rate at which the boxes are spawned are a fraction multiplied by the fuzzified GSR value. The Speed property was mapped to the convergence speed from the box's starting point to the participants' position in the world (**see Appendix: Section 1 – Methods: Visualisation Components**).

To summarise, the visualisation was composed of three main components:

- Participants' position within the Dome
- Boxes (Spawnable) instantiation class
- Mapping fuzzified data to the boxes

6.3.3.3 Background Colour Fader for Collisions and Contrasting

This script handled the background colour fader which traversed an array of five different colours for the background to gradually rotate between as the experiment unfolded. The colours used were such that there was a distinct colour difference, or contrast, between the boxes and the background (i.e. light boxes and dark background).

Another feature of the visualisation was the white-background fader which flashed a white screen for a second (including fade-in and fade-out effect) to indicate that a box has collided with the participant's position. (**see Appendix: Section 1 - Methodology: Unity3D Scripts – ThesisCamera.cs**).

6.3.3.4 Fetching JSON files from URL

This required the use of the System.Net library (Microsoft, 2016) and LitJSON (Leonardo B, 2016) third-party library (**see Section 7.5: Third-party libraries**). The Net library

conveniently provides a class that communicates with the FTP server using authentication whilst LitJSON was used for parsing JSON data.
The script was attached to the main camera of the scene such that it remained active throughout the duration of the experiment. This script handled the FTP connection and parsing of JSON data, and will eventually be stored in a class container for easy access at runtime (**see Appendix: Section 1 – Methods: Unity3D Program Scripts – GetURL.cs**). JSON data was stored in this class so as to easily make quick copies and new instances of objects with the participants' data.

This script worked by requesting access to a pre-defined URL in order to download the first file from the remote server. As previously explained in the **Data Processing Module: FTP Connector** section, files were uploaded with an index followed by '_fuzzyData.json'. Therefore it was easy to keep downloading from the web host by iterating over the index every 10 seconds. The reason for requesting URLs every 10 seconds was mainly due to the UI being blocked on download (**see Section 10: Limitations**).

In order to avoid the requested URL timing out or remaining in a loop of requesting the same URL, the index was incremented once the URL was not found and preventing redirection to another URL. This seemed to be the most viable option considering that the upload of files from the Data Processing Module occasionally lost packets in transit (**see Section 10: Limitations**).

The URL request utilised a coroutine function provided in the Unity Scripting API which essentially operated as an independent thread (Unity3D, 2016). Coroutines are convenient when downloading from a URL since it does not block the UI of the application while it downloads the content of the file. Rather it downloads all the content separately to the Update() thread and then merges it back with the file once the request is complete and successful (**see Appendix: Section 3 – Methods: Unity3D Program Scripts – GetURL.cs**).

Once the JSON file downloaded successfully from the URL, the index was then incremented, the relevant flags modified and the process repeated for the next URL. Files were added continuously from the start of the experiment, up until the moment the compilation of songs end (**see Section 3.4. Methodology – Experiments Figure 15: FTP authentication and JSON file downloader**).

6.3.3.5 Decoding and Storing JSON data

This part builds upon the previous sub-section - that is JSON files must be downloaded in order to start parsing data. As such, the parser method was invoked once the coroutine web request was successful, as described in the previous sub-section.
Once the web request was successful, the static method ReadJSON() was called to start parsing the content of the file and was ultimately stored in a class container utilising the Singleton pattern. This was done so as to keep the class accessible throughout the duration of the experiment (**see Appendix: Section 3 – Methods: JSON Parser and Storage of data in Singleton Class**).

LitJSON split up the JSON object into three main components:

- Object Start

- Property Name
- Object End

Object start indicates the starting (left curly bracket) of a JSON object, as previously described in the JSON encoder sub-section. Similarly, Object end indicates that the JSON object is closing (right curly bracket). Finally, the property name indicates what piece of data is coming next. Thus if the property name dictates 'pulse' then the following string to be read will be the value related to pulse.

The file was a JSON object containing 32 nest JSON objects.

The file was composed of a single large JSON object containing 32 nested JSON objects. The three main components provided by LitJSON required an algorithm which iterated over 32 rows in the file and repeated this process for the next files, while ensuring no data was lost (**see Appendix: Section 3 – Methods: Unity3D Scripts – GetURL.cs**).

6.3.3.6 Mapping data to boxes in Visualisation

The process of extracting 32 rows from a JSON file, stored in class container and the mapping of the values to the instantiation of the boxes was repeated every 10 seconds for the duration of the experiment. Therefore the process of mapping data onto boxes was applied in the exact manner for every other file, thus every other box is defined in the same manner.
It is worth noting that the fetching of JSON files had to be done before the instantiation of the first box in the visualisation. This was performed by utilising a convenient function provided by Unity, enabling the boxes to be instantiated with pre-defined properties (hathol, 2016).

Thus for every box instantiated, each of the four main properties were passed directly to the game object prefab (**see Appendix: Section 3 – Methods: Unity3D GameObjects and other utilities**), which were previously extracted from the class container that was continuously being populated, as described in the section **Fetching JSON files from URL**.

Since the Update() method in Unity is invoked 60 times per second, the mapping process required a function to act as a data buffer. JSON files were fetched from the URLs, and once downloaded – data was continuously stored in a class container. Finally boxes were instantiated with the decoded properties by extracting them from the class container row by row, file by file (**see Appendix: Section 3 – Methods: Mapping data for the visualisation**).

The reader is advised to refer to **Appendix: Section 1 – Methods: Unity3D snapshots** to see examples of the visualisation in action.

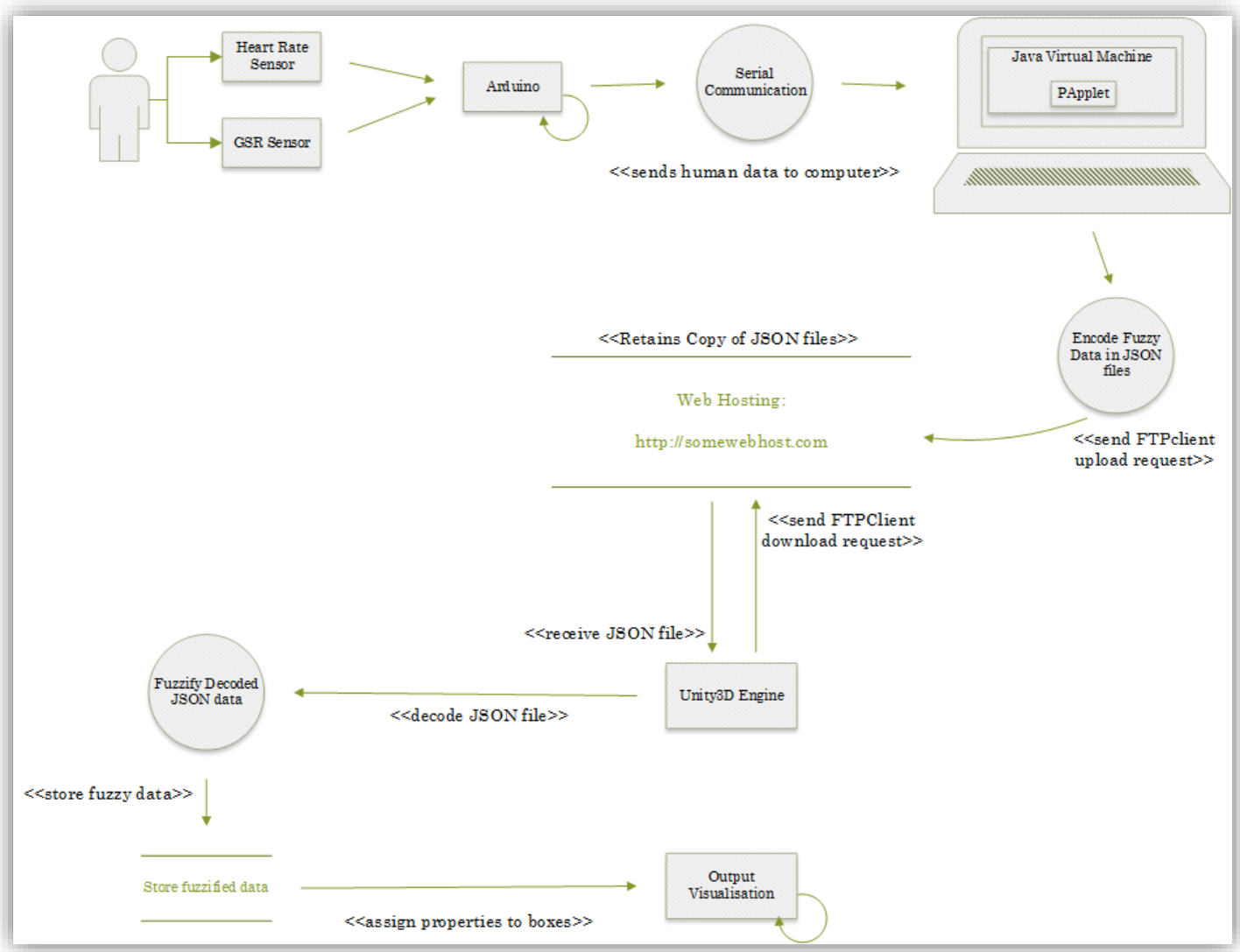


Figure 36: *The Input and Data Processing Modules.*

6.4 Conducting stability reading tests for GSR and HR sensors

Prior to conducting experiments, several stability tests were developed to deduce whether the incoming readings from the sensors were sufficient to be used in the experiments.

By following these instructions (Fustini, 2016) and (Villarejo, Zapirain, & Zorrilla, 2016), GSR signals were classified into regions that were used to map from GSR to Speed in the Fuzzy Logic System. Different resistors of varying resistance (measured in ohms) were used to test the resistance of the signal received when the participant comes in contact with the sensor (**see Appendix: Section 1 – Methods: Classifying GSR regions with resistors**). This was calculated using the principle that electrical conductance is the reciprocal of resistance. Thus the formula used to calculate the electrical conductance measured in Siemens (S) is:

$$\frac{1}{\Omega} \; where \; \Omega: \;\; 10\,\text{k}\Omega \leq \Omega \geq 10\,\text{M}\Omega, \qquad \Omega \text{ is Ohms}$$

This followed by measuring the skin conductance of individuals at a variety of ages, with varying skin conditions. For instance, one of the participants aged 58 worked as a Goldsmith, therefore his hands were pretty rough. This was an interesting factor to consider since the measurement of skin conductance (galvanic skin response) does not inherently depend on the skin condition, but rather how conductive or more resistant the participant is.

8 different people were tested in order to guarantee the stability of the readings:

- 2 of higher-bracket age (50+)
- 1 of lower-bracket age (30+)
- 5 of young-bracket age (20-29)

6.5 Testing jFuzzyLogic

As described in the Preliminary Project Report, jFuzzyLogic was tested with various inputs in order to test scalability when handling a lot of data at once. It was found that at least 600 simultaneous readings could be processed at any given time without much worry.

6.6 Third Party Libraries

- LitJSON for Unity3D (Leonardo B, 2016)
- DOTween for Unity3D (Giardini, 2016)
- Google Cardboard Library for VR headset (Google, 2016)
- Beads Audio Library for Processing (Bown, 2015)
- Processing IDE core.jar integrated in Eclipse (Java) (Foundation, 2016)
- RXTX Library (Fizzed, 2016) and JSSC (scream3r, 2016) for Serial Communication in Java
- jFuzzyLogic – Fuzzy Logic Controller in Java (Cingolani & Alcalá-Fdez, 2015)
- JSONObject for encoding JSON objects (org.json, 2016)
- Apache Commons for FTP requests (Apache, 2016)

6.7 EXPERIMENTS

6.7.1 Questionnaire Design

The questionnaire was designed to elicit the same responses from both the VR and traditional screen experiments. It was split up into four sections (demographics, first flag quiz, second flag quiz and experiment overview) which was partially completed before the experiment started and concluded once the visualisation ended.

The first section of the questionnaire focused on participant demographics and some basic information about their daily habits, alongside gauging an idea whether they know about the research field under study. The second section required participants to answer the first flag quiz.

The third section was filled in once the visualisation had been completed. This was used to measure participant performance against their performance in the second quiz, which started after the visualisation. The factors measured were response time, thought process, correctness and time taken to submit the final quiz.

The final section was used to qualitatively measure participants' response to the experiment by correlating two populations (VR and Traditional groups) with Principal Component Analysis statistic using IBM SPSS Statistics which can be seen in S**ection 7: Results** (IBM, 2016).

6.7.2 Experiment Procedure

The experiment required a standardised procedure so as to maintain consistency between participants. Each participant was to be tested in a quiet and controlled environment with no one else in the room but the author of this project (for observation and note taking).

Two groups of a similar mix (demographic characteristics) were chosen to investigate whether there exists a larger impact on sensory overload when considering more immersive-capable hardware. Ten participants were chosen to sit for the Virtual Reality experiment using the Google Cardboard headset while the other ten sat for the Traditional screen experiment using a laptop with a resolution of 1366x768. **See Section 7: Results** for more information about the experiment context.

The following points cover the necessary steps taken for every participant in order to ensure stable readings, while also guaranteeing the experiment did not result in error.

	STEPS TAKEN BEFORE PARTICIPANT ARRIVED:
1.	Ensure Arduino is set up appropriately
2.	Open Arduino IDE and upload code to Arduino Controller
3.	Open Processing IDE
4.	Ensure Internet Connection is stable
5.	Ensure available FTP port
6.	Ensure data is cleared from FTP client/web storage

Table 1: Steps to take for every experiment, prior to starting

STEPS TAKEN WITH PARTICIPANTS:

1.	Participant wears HR and GSR sensors
2.	Turn on Processing IDE for participants' benchmark sample
3.	Participant is informed about what the experiment entails
	a. Fill out first two sections of Questionnaire
	b. Explain:
	i. How data is stored, sent and received
	ii. Visualisation represents human and music data
	iii. Destroy boxes with left-mouse button
	iv. Let boxes hit you results in white flash
	v. Opt-out option at any point during experiment
	vi. Spraying Air Freshener
	vii. Possibility of Photos/Videos taken during experiment
	viii. Only author will be observing participant
	ix. Answer final two sections of Questionnaire
	x. Finally answer some more questions about the experience
4.	Process begins once music starts
	a. Request participant fill in first two sections of Questionnaire
	b. Participant wears headphones
	c. Refresh participants' minds about the experiment process
	d. Start transmitting data to web host
	e. Start visualisation which reads data from web host
	f. Spray Air Freshener throughout experiment
5.	Process ends once music stops

Table 2: Steps to take for every experiment when about to start experiment.

6.7.3 Participants chosen for the research project

A mixed (quantitative and qualitative) research approach **(Dawson. C, 2008, p. 47)** was adopted, wherein a small number of people were sampled for the experiments. This enabled the necessary data to be obtained for further analysis, correlations and to meet the defined objectives.

Although the heart rate and galvanic skin response readings vary from one participant to another, it is important to note that it does not matter since the elevation of levels is what I sought after. Participants subconsciously knew they are being measured, though they did not have any indication of what their current levels dictate. This was done so as to mitigate the measurement effect (Dewey, 2016).

The readings were used in combination with two other properties of sound, which together defined the four properties of the visualisation. The participants chosen were carefully selected in order to achieve a good mix of results considering similar demographic characteristics however their varying background could make for distinct differences. It was also imperative that the selection of participants differed in ages so as to better gauge whether the working background or previously experiences in life had an impact on their performance in the experiment.

There were a total of 20 participants that contributed to the project experiments. 10 were chosen to view the visualisation using a traditional screen while the remaining 10 experienced the visualisation in a virtual reality environment using a HMD.

The experiments took place in controlled environments which were not conducted at one location. This might have resulted in some issues relating to stability and potentially harm to the hardware devices see **Section 8: Limitations**.

AGE	VR	TRADITIONAL
< 18	1	0
20 - 24	6	4
25 – 29	2	1
30 – 39	0	2
>= 40	1	3

Table 3: Mix of participants for groups – VR and Traditional

The ideal participant choice would have consisted of 2 from each age category with at least two women for both the VR and Traditional experiment. However, due to strict time limitations and the inherent length of the experiment, not much could be done in terms of participant choice. This is further discussed in **Section 8: Limitations**.

6.8 JUSTIFICATION

6.8.1 Introduction

This section explains the justification for the different types of devices and approaches taken for the system and experiment to unfold. The input to the system is first discussed, followed by the output with reference to the visualisation.

6.8.1.1 Input

6.8.1.1.1 Arduino

Although there exist a variety of micro-controllers that can be used to elicit human data through sensors, given prior experience and the simplicity of data handling with Arduino, it seemed to be the ideal candidate to meet the pre-defined objectives. This concept was further supported by (jocelynzada, 2015) which visualised stress levels using Arduino as the medium for passing data.

6.8.1.1.2 GSR and HR Sensor

As previously described in **Section 4.3. System modules**, the GSR and HR sensors were used to elicit data from participants which was ultimately decided upon in the literature review stage in the preliminary stages of the project.

These two components allowed for the capture of relatively accurate stress and heart rate measurements, which although did vary from one participant to another, were deemed to be consistent in preliminary tests described in **Section 4.4: Conducting stability reading tests for GSR and HR sensors**.

Finally, I was able to elicit such data from participants and use it to correlate the impact of VR and Traditional environments with respect to the information sent to the human senses and sensory overload.

6.8.1.1.3 Headphones

The headphones were essential to the experiment since it further immerses the participant and isolates them. Having sound played from external speakers, the participant might find themselves influenced from various factors such as noise, low volume, interruptions/distractions and many more.

6.8.1.1.4 Data Processing

Given previous knowledge in Processing and Arduino, it was an obvious choice to implement my system using such software apart from the open-source libraries offered by other members in the community. The Arduino itself is simply a micro-controller

which supports multiple components and can be used to transmit electrical signals to the computer to be further interpreted.

6.8.1.1.5 jFuzzyLogic and PApplet in Eclipse

There are many public and private fuzzy logic systems that employ the fundamental concept of Fuzzy Logic however some are unclear or are simply overkill for my requirements.

Examples of private software includes MATLAB which has a variety of sub-components, one of them being Fuzzy Logic whereas an example of public fuzzy logic software include jfuzzylite (jcrada, 2015) and jFuzzyLogic (as described in various sections throughout this project).

Overall, jFuzzyLogic was the ideal choice for the job as it provided an easy-to-use eclipse plugin which allows the user to model a fuzzy logic system by simply defining inputs, outputs and rule blocks. The reader is advised to refer to the tutorials on how to use jFuzzyLogic on their website which shows their API documentation (Cingolani & Alcalá-Fdez, 2015).

6.8.1.1.6 FTPClient

This class which is free to the public in the Apache Commons provides methods to communicate with the FTP server that was created for this project to enable live-sync of human data to visualisation output.

In order to do so, I registered for a free-hosting service that will house my data (000webhost, 2016). Once registered, I downloaded and setup FileZilla (FileZilla, 2016) to establish an FTP connection and upload/download files for the experiment.

Once the experiment starts, the data processing module continuously fetches data and stores it in a buffer and uploads said data in batches using JSON Objects (org.json, 2016) and thus stored in '.json' format.

6.8.1.2 Output

6.8.1.2.1 Laptop Screen

To standardise the experimental process, I opted to use my laptop screen for the visualisation with respect to the traditional experiment participants. This choice was desirable due to portability issues when having to conduct experiments at different locations.

Other benefits for the laptop screen over other options such as monitors include the lack of required electricity to power on, monitors are bulky and the experiment sufficed from the laptop screen.

6.8.1.2.2 Google Cardboard VR

The use of Google Cardboard (**GC**) was decided way back in July 2015 due to high requirements implied by the Oculus rift. Such requirements made it difficult for the experiments portable and thus, considering that GC is a wireless VR headset, experiments could essentially be performed anywhere.

The only restriction is that of electricity required to maintain laptop battery charge. To remain consistent with every experiment, each participant was tested and observed using the same technology, same sensors, and same stimulus except for the environment in which the participant was tested and observed in.
GC was provided for VR participants which were to experience a series of stimulus through four senses.

Participants could interact with the visualisation by destroying the boxes on screen by pulling the trigger button on the left side of GC. The destruction of boxes was made purely option for the participants.

6.8.1.2.3 Mouse

On the other hand, traditional experiment participants were provided with a mouse interact with the visualisation. Likewise, the destruction of boxes was made optional for the participants. This form of interaction expands the level of immersion that participants experience since they could use it to pass time or control their level of sensory overload.

6.8.1.2.4 Unity3D

Although animation is made easy in Processing, Unity3D has its own perks. Given the in-built physics engine in the free-version of Unity, one cannot resist designing a fully-fledged animation without the need to design the animation requirements.

This engine allowed me to create a dynamic visualisation which spawns a series of boxes that constantly move towards the viewer (participant) position on screen. Unity also has the advantage to build to android devices, thus further allowing me to design a one-for-all visualisation that catered for both VR and Traditional experiments.

7 RESULTS

7.1 INTRODUCTION:

The first part of this section gives context to the location and equipment used for the experiments while the rest of the section a detailed analysis of both quantitative and qualitative results gathered throughout the experiments is presented. The main comparisons are made qualitatively however a comparison of quantitative data is made in relation to the group of traditional participants.

All the necessary data pertaining to the results in this section can be found in the **Appendix: Section 3 – Results jargon**.

7.2 EXPERIMENTS:

7.2.1 **Location:**

Participants were tested in four different locations where each participant was placed in a controlled environment which consisted of no distractions, sound or otherwise. Similarly, the room in which the participant was examined only consisted of the observer (author of this project) and the participants themselves. Photos and videos for the experimentation process may be found in the archive structure in the folder: Snapshots and Videos under the category *Multimedia*.

7.2.2 **Equipment:**

Several devices and equipment were required to conduct such experiments. This involved the Arduino micro-controller, individual heart rate and galvanic skin response sensors, Google Cardboard (HMD), a laptop (TDE), a mouse, a pair of headphones and air freshener. The brand used was Febreeze air freshener which was used to stimulate the participants' smelling sense.

The laptop used was the Fujitsu A531 Lifebook of screen width 17.3" and resolution of 1366x768, while the mouse simply consisted of a basic Logitech brand with laser movement. An Arudino-UNO was used to connect the Heart Rate and Galvanic Skin Response sensors. These sensors were built by SeeedStudio in the U.K. The mobile used for HMD consisted of a OnePlus One 64 GB Sandstone Edition

Each participant was given a brief explanation of the experiment process detailing the purpose for the participants' contribution, what is being measured and for what reason. They were also given a quick run-down of which tasks were to be completed from the start of the experiment till either the experiment terminated or termination at their own discretion. For reasons pertaining to sensory overload, participants were given the option to opt-out from the experiment at any moment they deem necessary.

Data obtained from the questionnaires may be found in the archive since they were too long to add into this document. Verbal response and observations noted mid-experiment may be found in the **Appendix: Section 3 – Observations and verbal reports**.

7.3 DEMOGRAPHICS

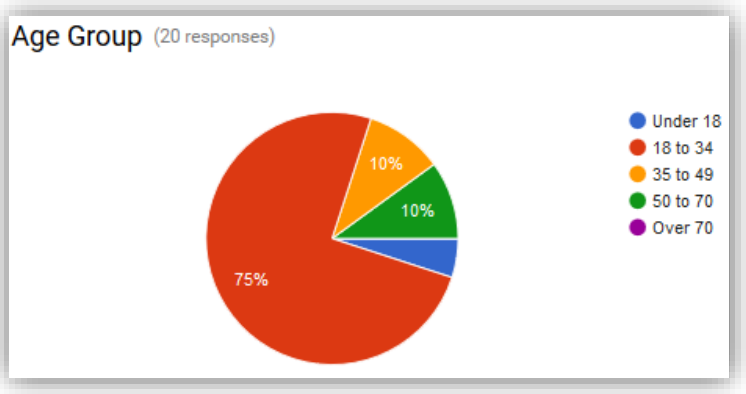


Figure 37: *Pie chart representing participants' age groups*

The pie chart above clearly shows that the majority of participants were aged between 18 and 34. One minor was tested for the Virtual Reality experiment however I could not find another minor for the Traditional experiment. The minor was of age 17 studying for his A-Levels in Malta.

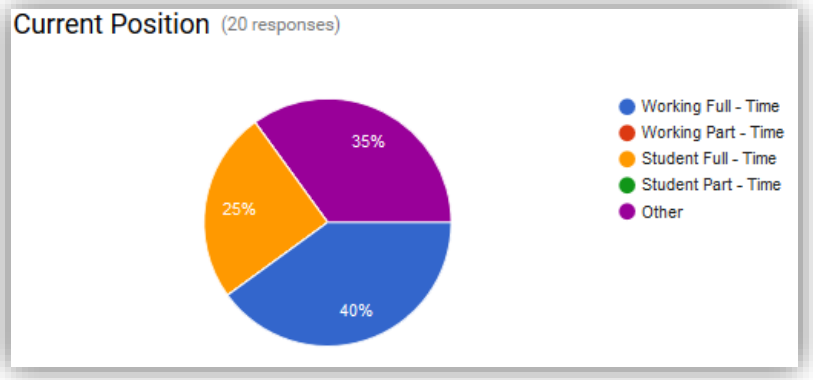


Figure 38: *Pie chart showing participants' current occupation*

In the case of current occupation, there seemed to be three distinct categories with the majority being working full-time. The option *other* included those who are both currently studying and working part-time, and one participant who is on early retirement but working part-time.

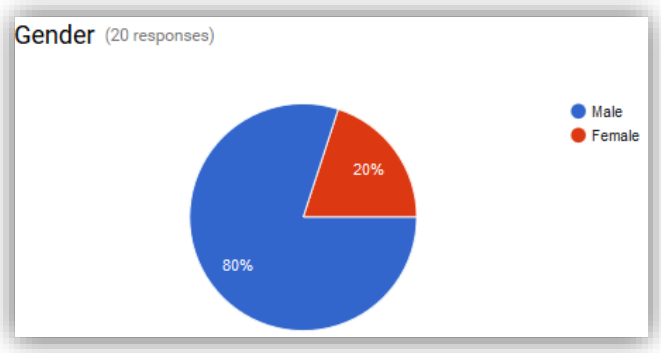


Figure 39: *Pie chart representing gender distribution*

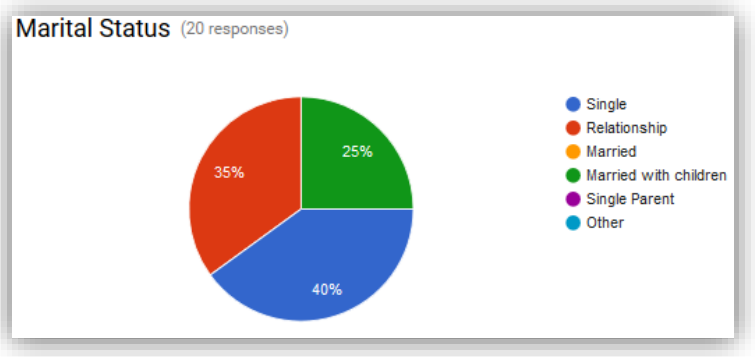


Figure 40: *Pie chart showing the participants' marital status*

As can be seen in the **Figure 39** above, out of 20 participants 16 were male and 4 were female. **Figure 40** shows three categories with the majority of participants being single. The other two categories are relationship and married with children.

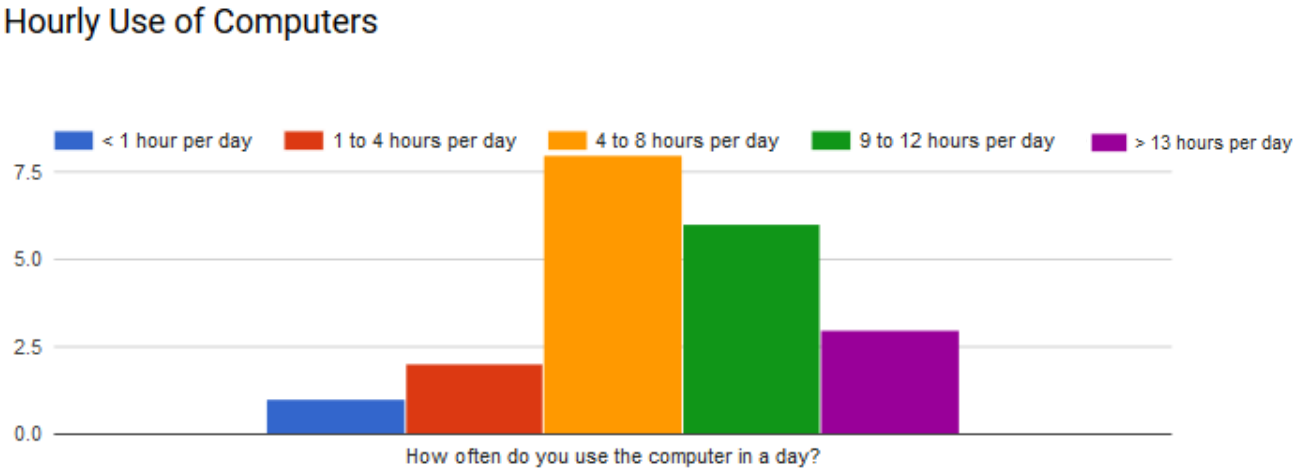


Figure 41: *Bar chart representing participants' daily computer use*

The majority of participants from **Figure 41** above appear to use computers for relatively long periods throughout the day with the majority of them lying within the 4 to 8 hour range. One participant used a computer for less than one hour per day while another two participants used it for 1 to 4 hours per day. There also are two participants that use a computer for more than 13 hours per day.

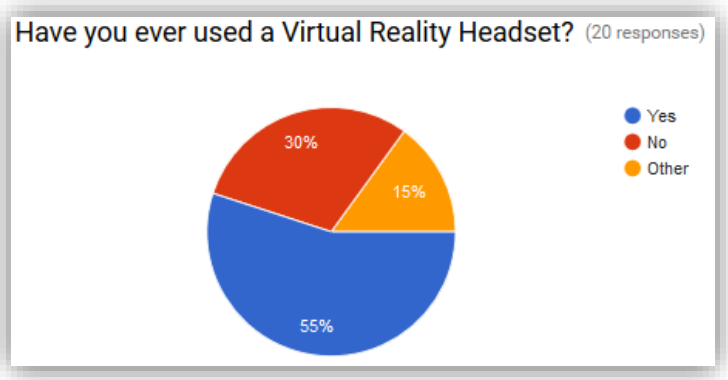


Figure 42: *Pie chart representing participants' prior experience with Virtual Reality headsets*

Figure 42 shows a majority of participants having used a Virtual Reality headset before, be it wired or wireless. Out of 20 participants, 6 had not used any form of Virtual Reality headsets and 3 had some form of interaction with it.

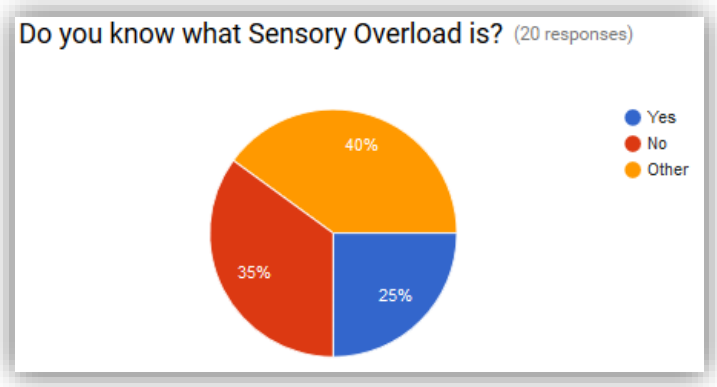


Figure 43: *Pie chart representing the distribution of participants' knowledge of Sensory Overload*

Figure 44: *Pie chart representing the distribution of those who knew the symptoms of sensory overload*

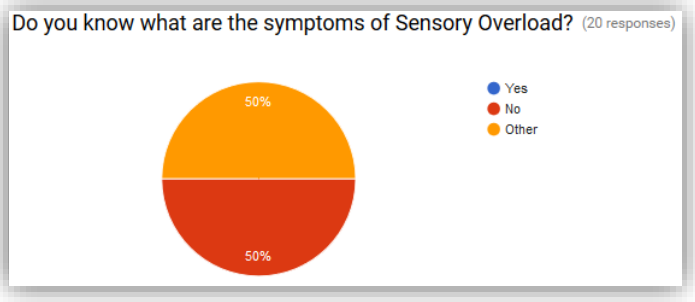


Figure 43 shows a majority of participants having a vague idea about what Sensory Overload is. **Figure 44** shows an equal distribution of participants who had any knowledge about the symptoms of sensory overload. Overall, few participants actually knew what sensory overload is, what it entails and how to recognise it.

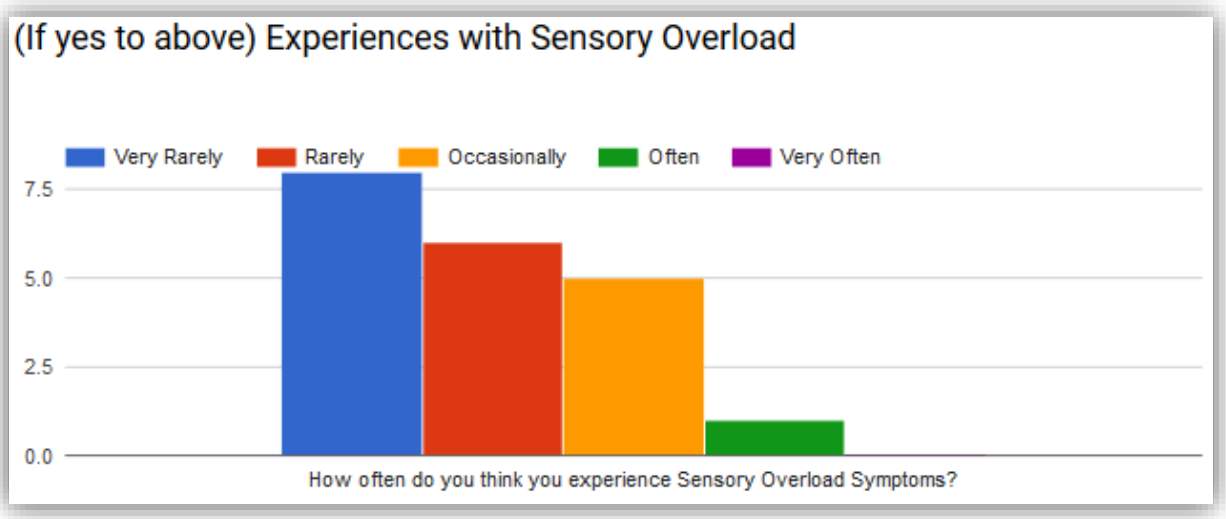


Figure 45: *Bar chart showing the number of participants' that realise when they experience sensory overload*

As shown in **Figure 45**, the majority of participants believe that they rarely experience sensory overload or at the very least, its symptoms. Those participants that knew what sensory overload is chose either rarely, occasionally and often with the exception of one participant who still believes that they rarely experience such symptoms.

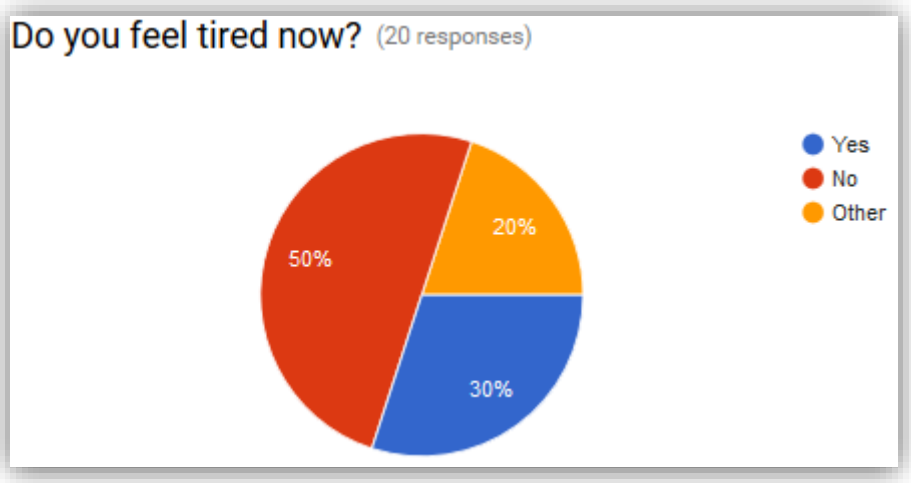


Figure 46: *Pie chart representing the participants' current fatigue, prior to the start of the visualisation*

Figure 46 shows a majority of participants starting the experiment not tired. Out of 20 participants, 6 of them were tired while the other 4 were either slightly tired from some previous activity or from work.

7.4 OVERALL EXPERIENCE

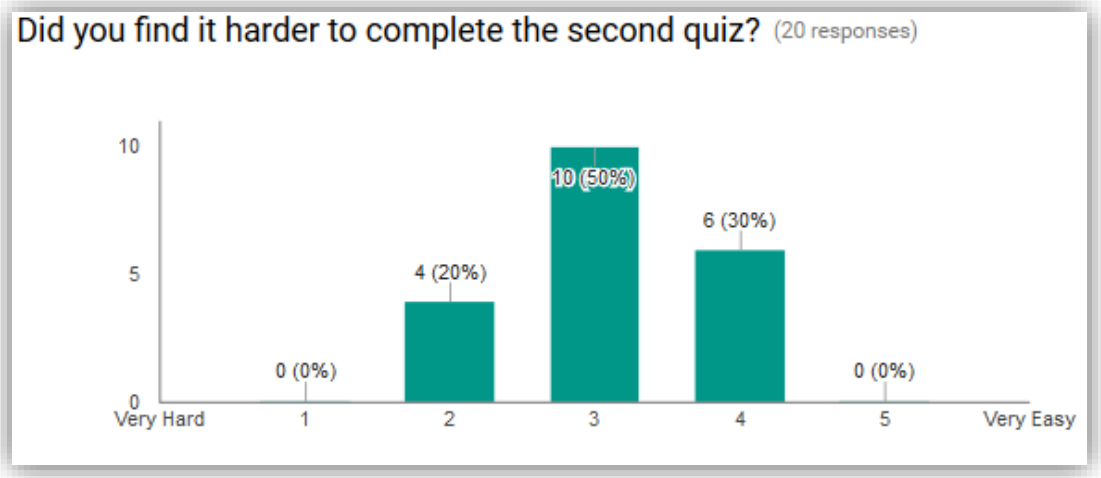


Figure 47: Bar Chart representing the difficulty in completing the second flag quiz in comparison to the first.

Figure 47 above shows a comparison of difficulty for 20 participants when completing the second flag quiz, after concluding the visualisation stage in the experiment. The majority of participants found the second quiz to be equal in difficulty to the first flag quiz. Six of the participants found it easier to complete the second quiz the other four found it more difficult to complete.

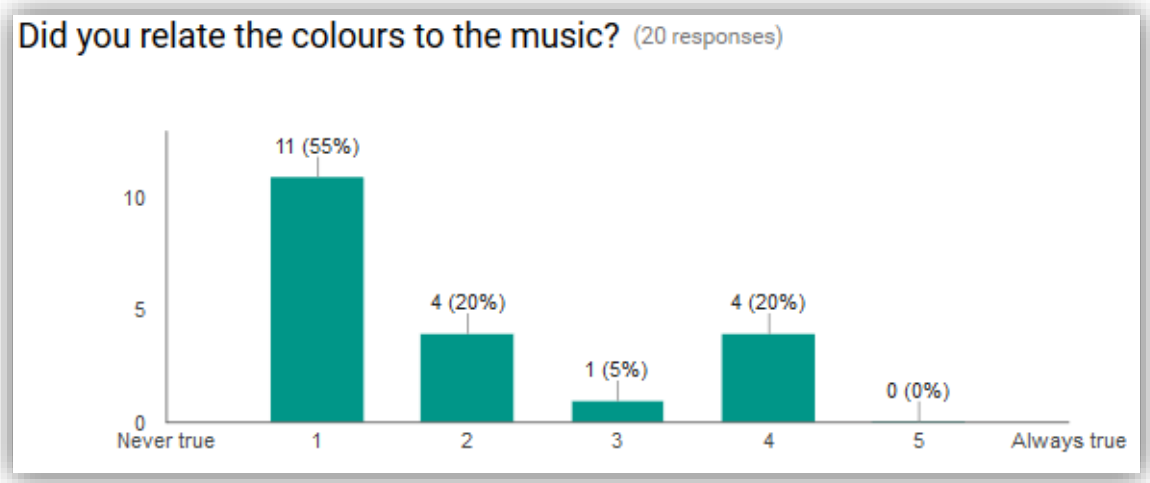


Figure 48: Bar chart representing the participants that saw a relation between the colours in the visualisation and the music

The chart in **Figure 48** above shows the number of participants that related the colours shown in the visualisation to the music being heard throughout the experiment. Four participants saw a close relation, four others did not think they saw a relation while one participant was not sure.

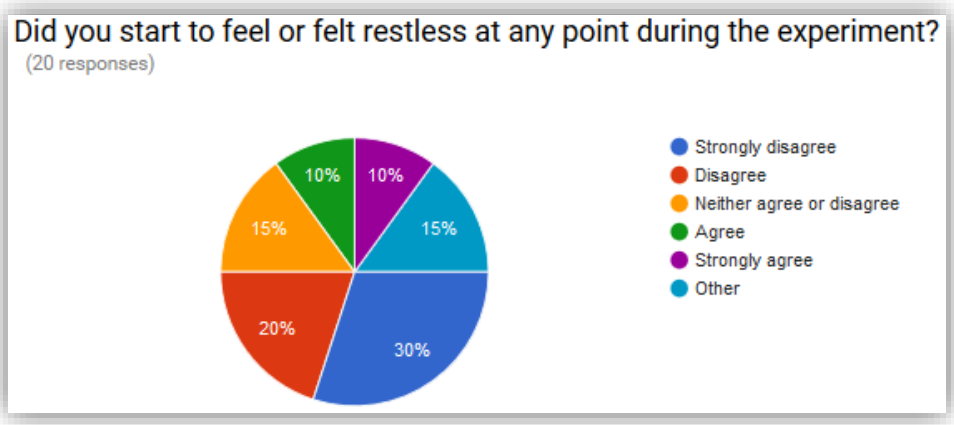


Figure 49: *Pie chart representing the participants who did or did not feel restless at any point during the experiment.*

The majority of participants in **Figure 49** did not feel restless what so ever. Out of 20 participants, 4 of them were passive and unbothered. Three participants were unsure while another 3 seemed to be irritated by something, but not essentially restless. Finally, two participants felt slightly restless or irritated while the last two participants felt very restless and irritated.

Figure 50: *Participants listed in chronological order providing their response in relation to Figure 19 above.*

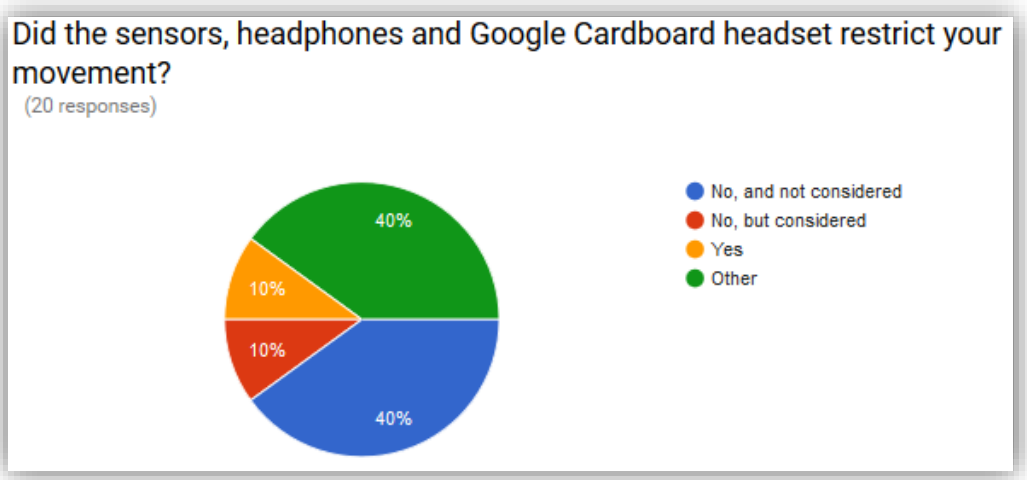


Figure 51: *Pie chart representing those participants that found the equipment restricting their movement throughout the experiment.*

Figure 51 above shows an equal distribution of participants that gave their own opinion about the equipment restriction and those who did not find it restrictive throughout the experiment. Two participants found the equipment to be restrictive with respect to their movement while another two did not find it to be restrictive but considered it several times.

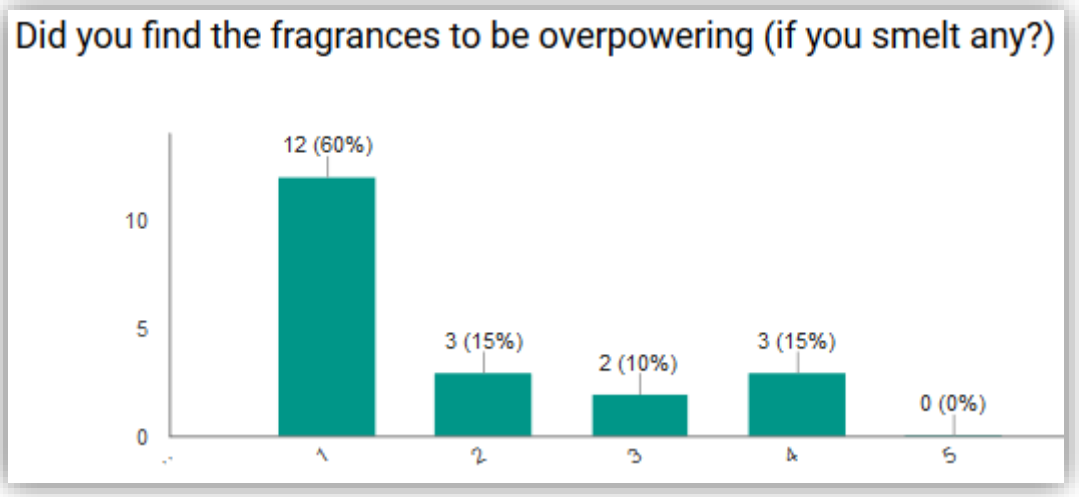


Figure 52: *Bar chart representing those participants that responded to the fragrance*

The majority of participants in **Figure 52** found the fragrances throughout the experiment to be pleasant. Three participants were not bothered by the smell while another three found the smell to be slightly overpowering. The last two participants found it to be only slightly overpowering.

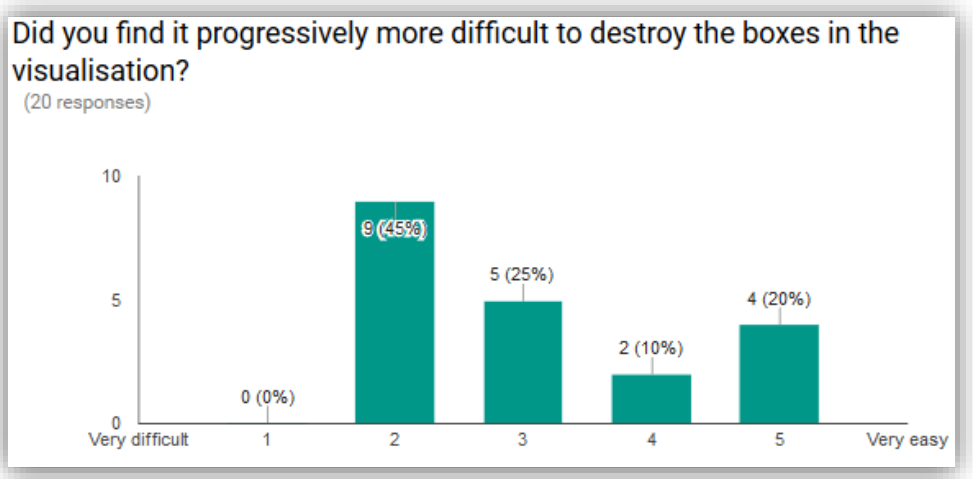


Figure 53: *Bar chart representing the participants' response to the visualisation boxes.*

Figure 53 shows the majority of participants' destroying of the boxes to be more difficult as time went on throughout the experiment. Five participants found it to be the same difficulty all throughout while four others found it very easy to keep destroying boxes. The last two participants did not find it very easy but still easy to a certain degree.

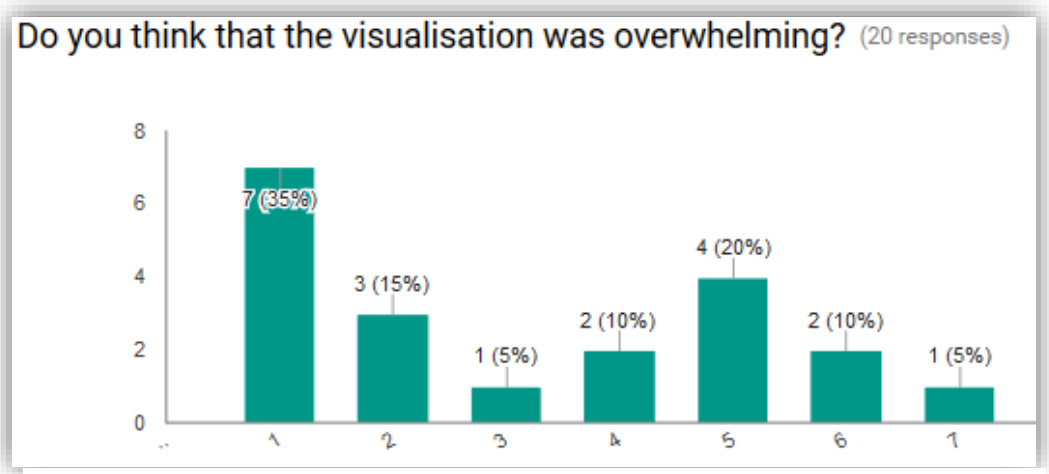


Figure 54: *Bar chart showing the distribution of participants' responses to the overall visualisation experience. '1' refers to Strongly Disagree while '7' refers to Strongly Agree.*

Figure 54 shows a larger distribution of participants indicating differing opinions about the overwhelming factor of the visualisation. The majority of participants did not find the visualisation to be overwhelming at all while four participants found it to be relatively overwhelming. Three participants did not feel overwhelmed (option 2), two were unsure (option 4) while two others were quite overwhelmed (option 6). One participant started to get disoriented while the remaining participant found the visualisation to be very overwhelming.

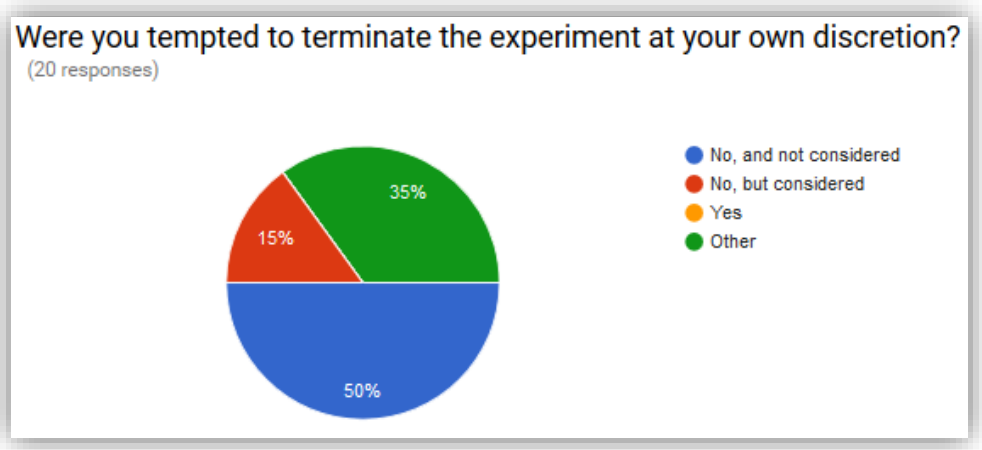


Figure 55: *Pie chart representing those participants who did not consider, those who considered and those who had conflicting arguments for terminating the experiment.*

The pie chart in **Figure 55** above shows a majority of participants that did not chose to and neither considered terminating the experiment. Seven participants had conflicting thoughts about terminating while the remaining participants did not want to terminate but considered the idea.

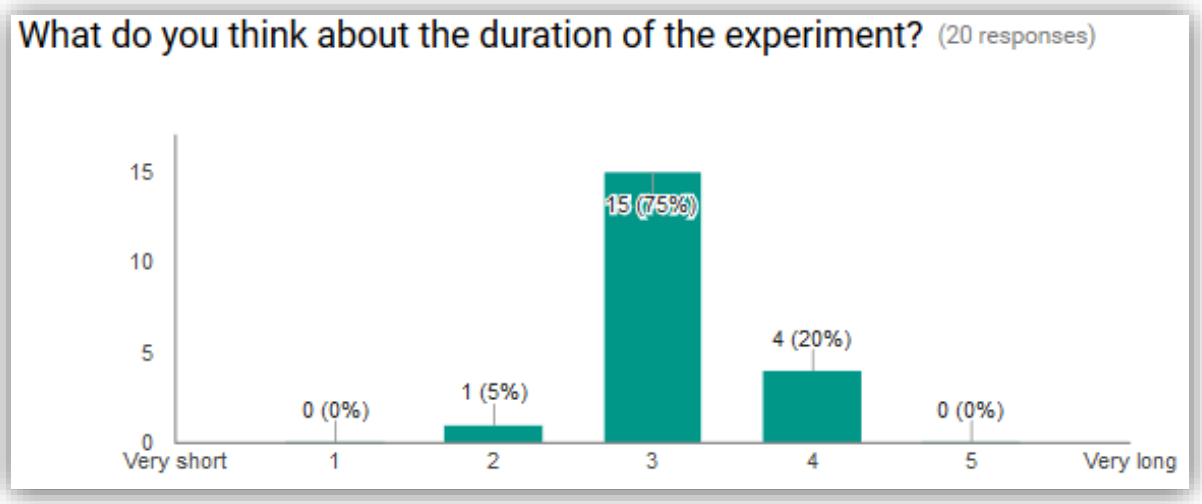


Figure 56: *Bar chart representing participants' response to the duration of the experiment.*

Figure 56 shows a majority of participants that felt the experiment to be perfect choice of timing while four participants found it to be slightly long. One participant that took part in the Virtual Reality experiment found the experiment to be slightly short.

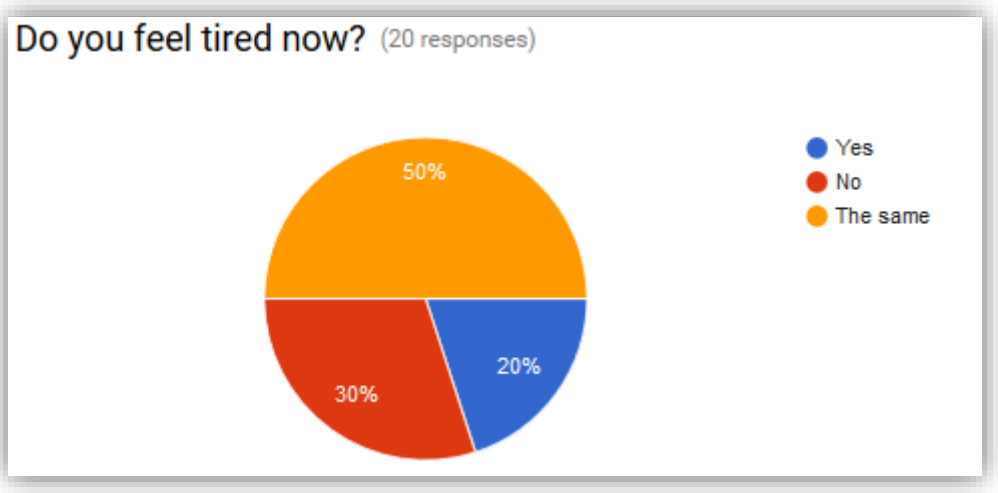


Figure 57: *Pie chart showing a comparison of participants' fatigue to the initial question in Section 1 of Results.*

Figure 57 above shows the majority of participants that felt the same in comparison to the start of the experiment. Six participants felt more alert while the remaining four participants felt more tired after the experiment.

7.5 QUALITATIVE DATA:

This data was collected using naturalistic observation, performing protocol analysis while also saving a copy of the participants' data to file. As explained in **Section 5: Methods**, the participants sent their heart rate and galvanic skin response data to the computer. This was further processed in the Fuzzy Logic Controller, wrapped into a JSON object alongside the components extracted from the music and stored on the web hosting service via FTP.

7.5.1 **Verbal Response**

Participants' responses varied slightly from one another however mostly responded on similar terms when asked structured questions. One particular question which seemed to show an equal distribution of response is that of whether they observed any relation to the speed and pulse rate of the visualisation.

Figures 57 and **58** show two bar charts which represent both groups, VR and Traditional participants. Participants responded similarly in terms of what they related. Out of 20 participants, seven appeared to relate nothing and simply focused on the visualisation rather than paying attention to such details. Six participants related both the pulse rate (heart rate) and the speed (galvanic skin response). Another six participants mainly related the speed at which the boxes moved towards the viewer (participant) during the visualisation. The last participant only related the pulse rate during the experiment.

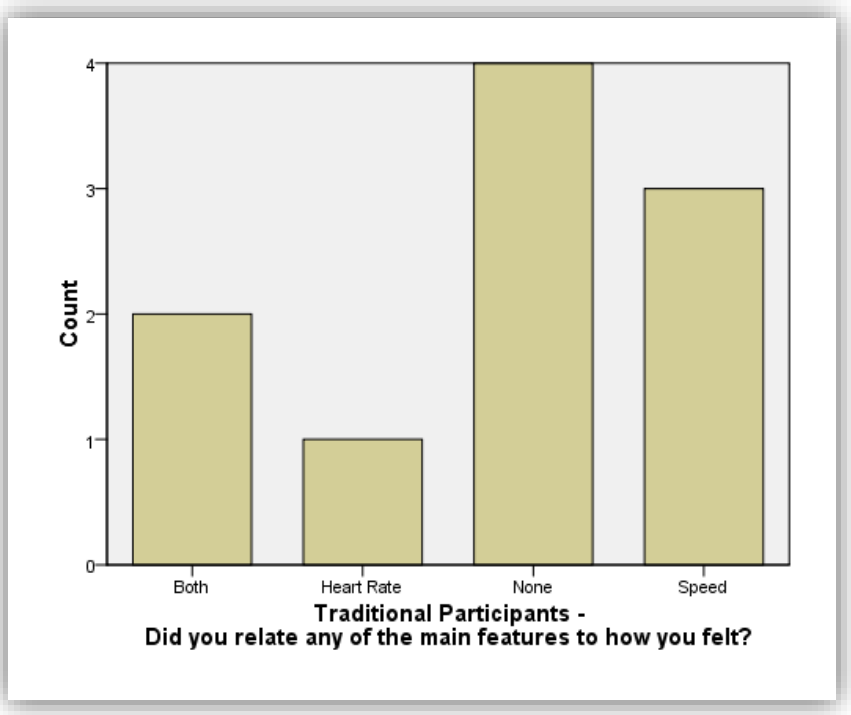


Figure 59: *Verbal response of Traditional participants' that realised relation in the visualisation.*

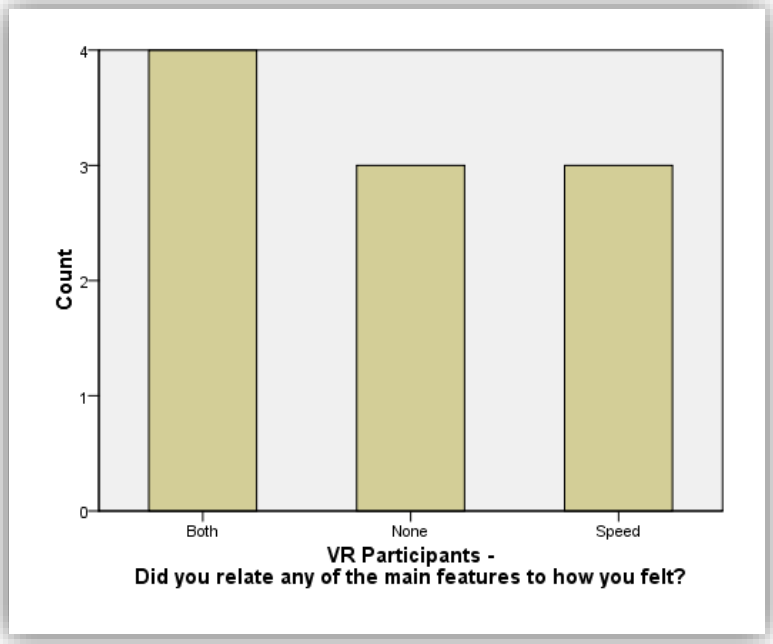


Figure 58: *Verbal response of VR participants' that realised relation in the visualisation*

The following tables represent all the comments obtained during the discussion post-visualisation and questionnaire for both VR and Traditional participants. The first table represents the additional comments reported by participants when using the VR display.

Additional Commments by VR participants	# of Participants commented
Screen rotated too much	**3**
Is there a relation?	**1**
Not centered anymore	**1**
They look similar & orientation is off	**1**
Restrictive colour palette	**1**
'Queen stopped' and Screen Panning	**1**
Nothing is showing anymore	**1**
Camera pans too much	**1**

Additional Commments	# of Participants commented
The faster I clicked, the more boxes came towards me	**3**
I realised that if I rushed, more come. If I slowed, they slowed down.	**2**
The pulsing was chaotic, felt somewhat similar	**1**
As I cleared the screen, everything started to calm down	**1**
The more I wanted to clear, the crazier it became	**1**
Not too sure	**2**
No relation, wasn't focused on that	**7**

Figure 60 on the following page represents the correlations between both groups of participants. In this case, the table represents the correlation based on GSR gathered from the participant. On the other hand, the pulse data obtained from VR participants did not sufficiently record appropriate HR data. This was probably due to movement with the HMD since data changes when the HR sensor cable is moved often. Thus no representation and comparisons by HR will be made. Further analysis of these GSR correlations made using Principal Component Analysis statistic can be seen in **Figures 61** to **65**, inclusive.

Reproduced Correlations

		Traditional - Participant 1	Traditional - Participant 2	Traditional - Participant 3	Traditional - Participant 4	Traditional - Participant 5	Traditional - Participant 6	Traditional - Participant 7	Traditional - Participant 8	Traditional - Participant 9	Traditional - Participant 10
Reproduced Correlation	Traditional - Participant 1	.791[a]	-.159	.565	.704	-.077	.295	.143	.144	-.480	.501
	Traditional - Participant 2	-.159	.916[a]	-.023	-.023	.920	-.066	.190	-.736	.124	-.048
	Traditional - Participant 3	.565	-.023	.893[a]	.797	.186	.197	-.481	-.308	-.181	.463
	Traditional - Participant 4	.704	-.023	.797	.908[a]	.131	.387	-.162	-.196	-.324	.530
	Traditional - Participant 5	-.077	.920	.186	.131	.986[a]	-.030	.012	-.851	.137	.072
	Traditional - Participant 6	.295	-.066	.197	.387	-.030	.447[a]	.118	.017	-.031	.319
	Traditional - Participant 7	.143	.190	-.481	-.162	.012	.118	.871[a]	.284	-.270	-.052
	Traditional - Participant 8	.144	-.736	-.308	-.196	-.851	.017	.284	.866[a]	-.219	-.087
	Traditional - Participant 9	-.480	.124	-.181	-.324	.137	-.031	-.270	-.219	.487[a]	-.198
	Traditional - Participant 10	.501	-.048	.463	.530	.072	.319	-.052	-.087	-.198	.533[a]
	VR - Participant 1	.530	-.006	.664	.517	.147	.132	-.248	-.106	-.106	.374
	VR - Participant 2	-.336	-.022	-.237	-.184	-.030	.173	-.080	-.084	.315	-.014
	VR - Participant 3	-.130	.514	-.082	-.180	.539	-.227	.050	-.434	-.005	.069
	VR - Participant 4	-.112	.925	.156	.096	.985	-.043	.013	-.850	.158	.043
	VR - Participant 5	-.096	.924	.165	.110	.986	-.035	.019	-.848	.149	.054
	VR - Participant 6	.392	-.004	.545	.637	.083	.315	-.148	-.151	-.102	.212
	VR - Participant 7	.037	.055	.158	.192	.081	-.151	-.148	-.199	-.251	.012
	VR - Participant 8	.652	-.039	.743	.810	.133	.389	-.199	-.202	-.254	.611
	VR - Participant 9	.088	.068	.298	-.032	.159	-.344	-.330	-.111	-.006	-.004
	VR - Participant 10	.249	-.048	.250	.293	.062	.335	-.093	-.115	.012	.502

Extraction Method: Principal Component Analysis.

a. Reproduced communalities

b. Residuals are computed between observed and reproduced correlations. There are 53 (27.0%) nonredundant residuals with absolute values greater than 0.05.

Figure 60: *Reproduced Correlation Matrix for raw GSR data using Principal Component analysis.*

7.5.2 Participant GSR Correlations:

7.5.2.1 High Positive Correlation

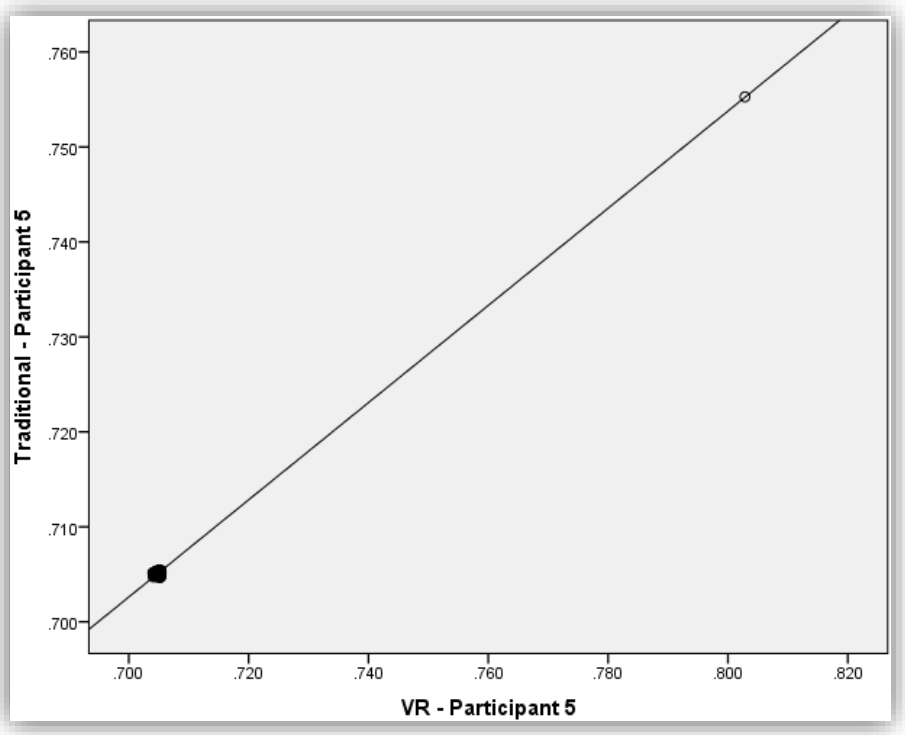


Figure 61: *High-positive correlation between Traditional Participant 5 and VR Participant 5*

7.5.2.2 Low-Positive Correlation

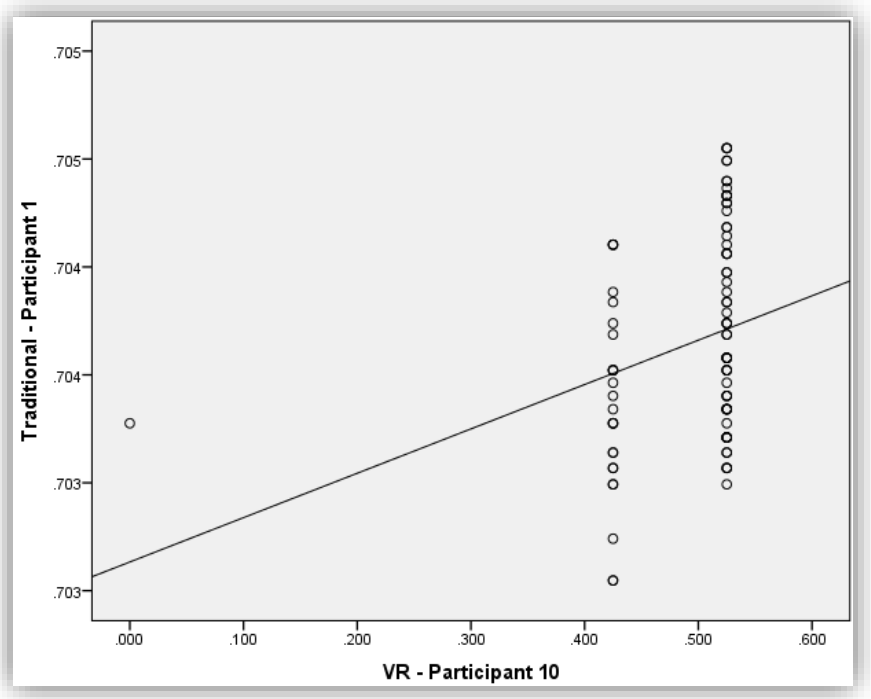


Figure 62: *Low-positive correlation between Traditional Participant 1 and VR Participant 10*

7.5.2.3 No Correlation

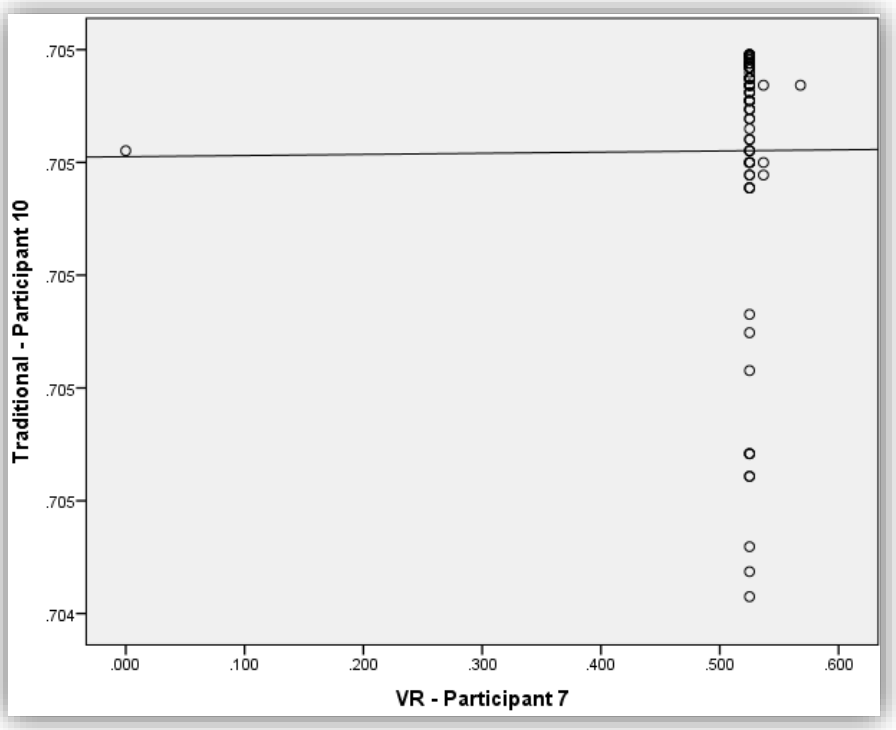


Figure 63: *No correlation at all between Traditional Participant 10 and VR Participant 7*

7.5.2.4 Low-Negative Correlation

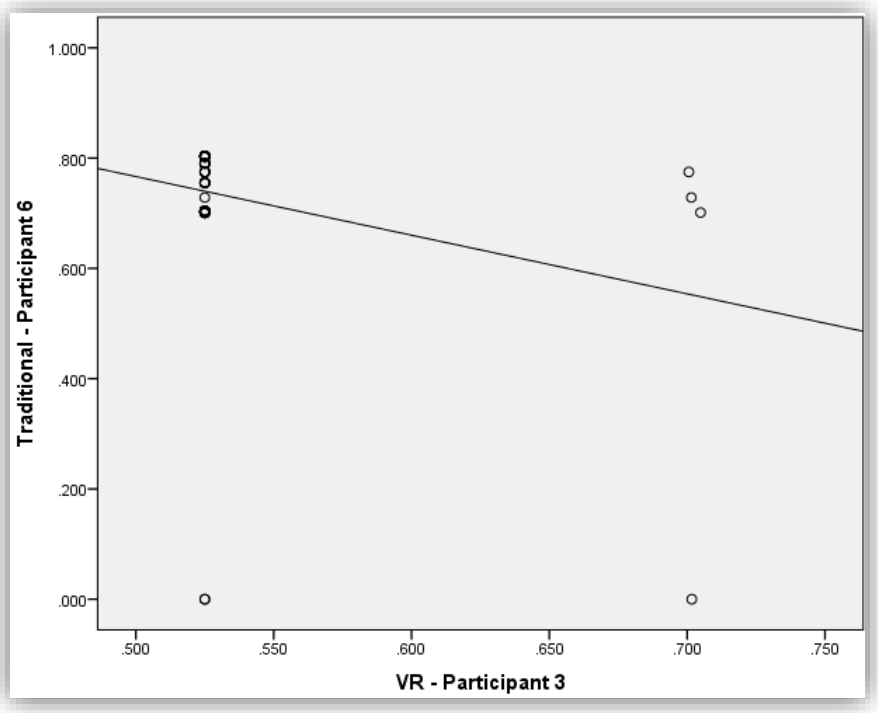


Figure 64: *Low-negative correlation between Traditional Participant 6 and VR Participant 3*

7.5.2.5 High Negative Correlation

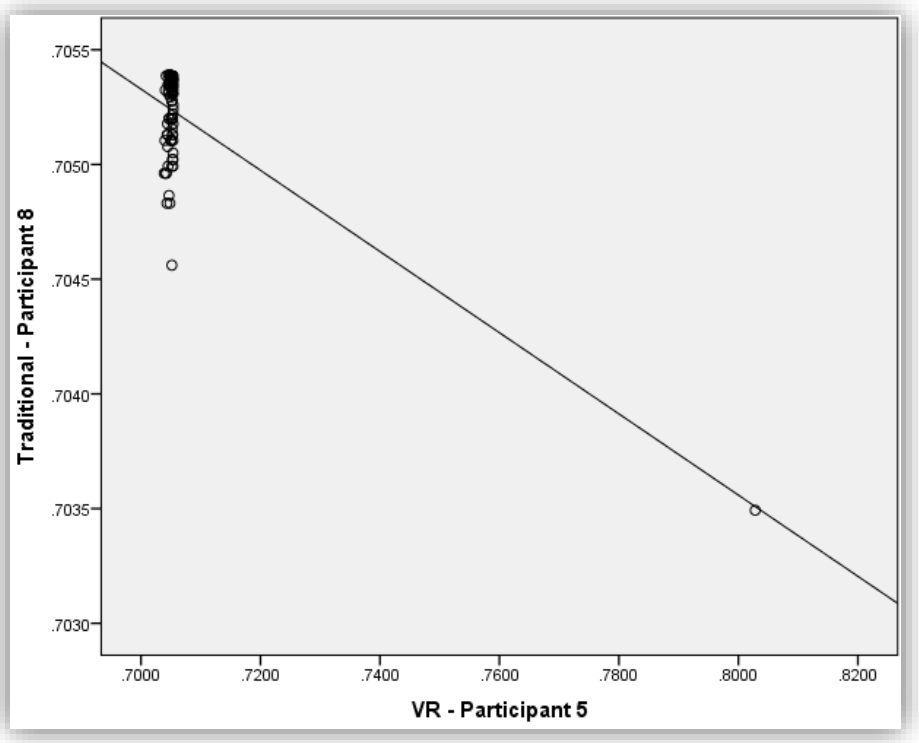


Figure 65: *High-negative correlation between Traditional Participant 8 and VR Participant 5*

Figures 61 through **65** represent the highest and lowest positive, highest and lowest negative and no correlations between participants from both groups. Positive correlation is denoted as a line starting from the origin or 0 and moves upwards to the top-right corner.

Negative correlations are the exact opposite to the positive ones in which the line starts at the very end of the chart and intersects with the y-axis at some established position. As can be seen from the figures 46 through 50, those that are both positive and negatively correlated see clusters of data points at some point on the chart. The chart depicting no correlations results in data points spread out on the chart while the trend line seems to be relatively straight.

It is interesting to note that those with low-positive and low-negative correlations contain both clustered and spread out data points which seems to align with the fact that they are somewhat correlated, however not consistently so.

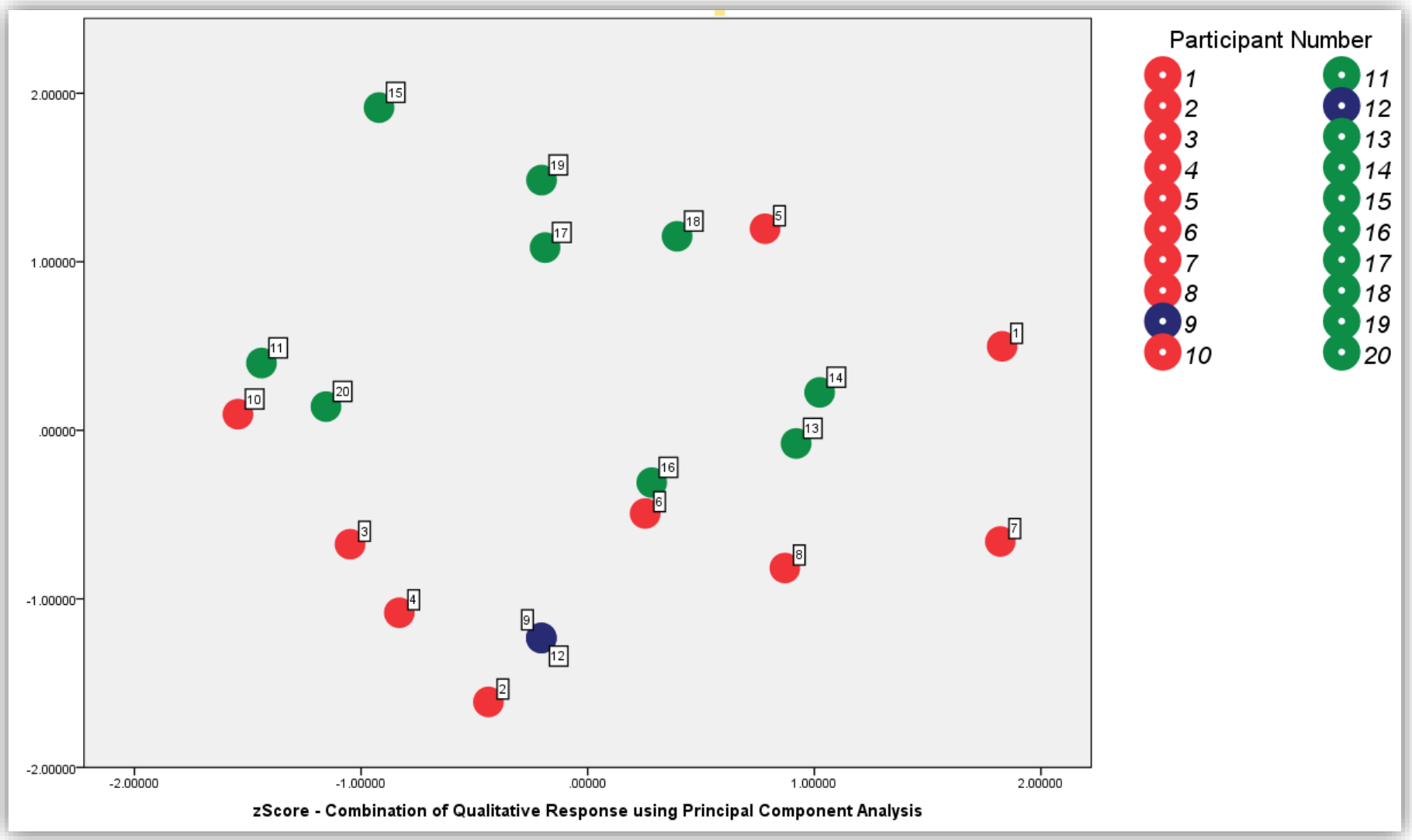


Figure 66: *Scatter plot representing VR vs. Traditional participants. Notice the separation, clustering and overlapping of participants. VR participants denoted as red while Traditional participants denoted as green.*

Figure 66 shows a scatter plot with each data point representing a participant. Each participant is further defined by five dimensions - distraction, duration, overwhelming, restricted and restless. These dimensions were obtained through qualitative questions during the experiment.

The green points represent the traditional group of participants while the red points represent the VR group of participants. The blue data point signifies overlapping participants. In this case it appeared that participant 9 of the VR group and participant 2 of the traditional group seemed to have a high-positive correlation together. According to the qualitative data gathered from the questionnaires and the plot in **Figure 66**, it appears that the closer participants are to each other, the less sensory overload related symptoms were exhibited. On the other hand, the more isolated participants generally scored higher in exhibiting sensory overload symptoms.

Correlation Matrix

		Difficulty	Duration	Overwhelming	Restricted	Restless
Sig. (1-tailed)	Difficulty		.206	.181	.079	.016
	Duration	.206		.073	.077	.059
	Overwhelming	.181	.073		.129	.264
	Restricted	.079	.077	.129		.317
	Restless	.016	.059	.264	.317	

Table 4: *The correlation matrix representing the scatter plot in Figure 51.*

As can be seen in **Table 4** above, the highest positive correlation exists between the Restricted and Restless dimensions, followed by Overwhelming with Restless. The third highest is Difficulty with Duration which shares the same value in terms of correlation with each other. The closer the value is to 0.000, the lesser the correlation that exists between such dimensions.

The following **Figures 67**, **68** and **69** represent individual groups for direct comparison amongst each other. Considering the clustering of participants in **Figure 55**, which could be further clustered into smaller groups indicates a sense of similarity amongst participants that were independent of each other.

Figure 67 represents the group of participants using a HMD during the experiment. This chart shows few participants clustered together and thus relate to each other in some manner. Participants 1 and 5 appear to be mostly isolated from the rest of the group lending to the idea that they must also be related to each other, regardless of their distance apart.

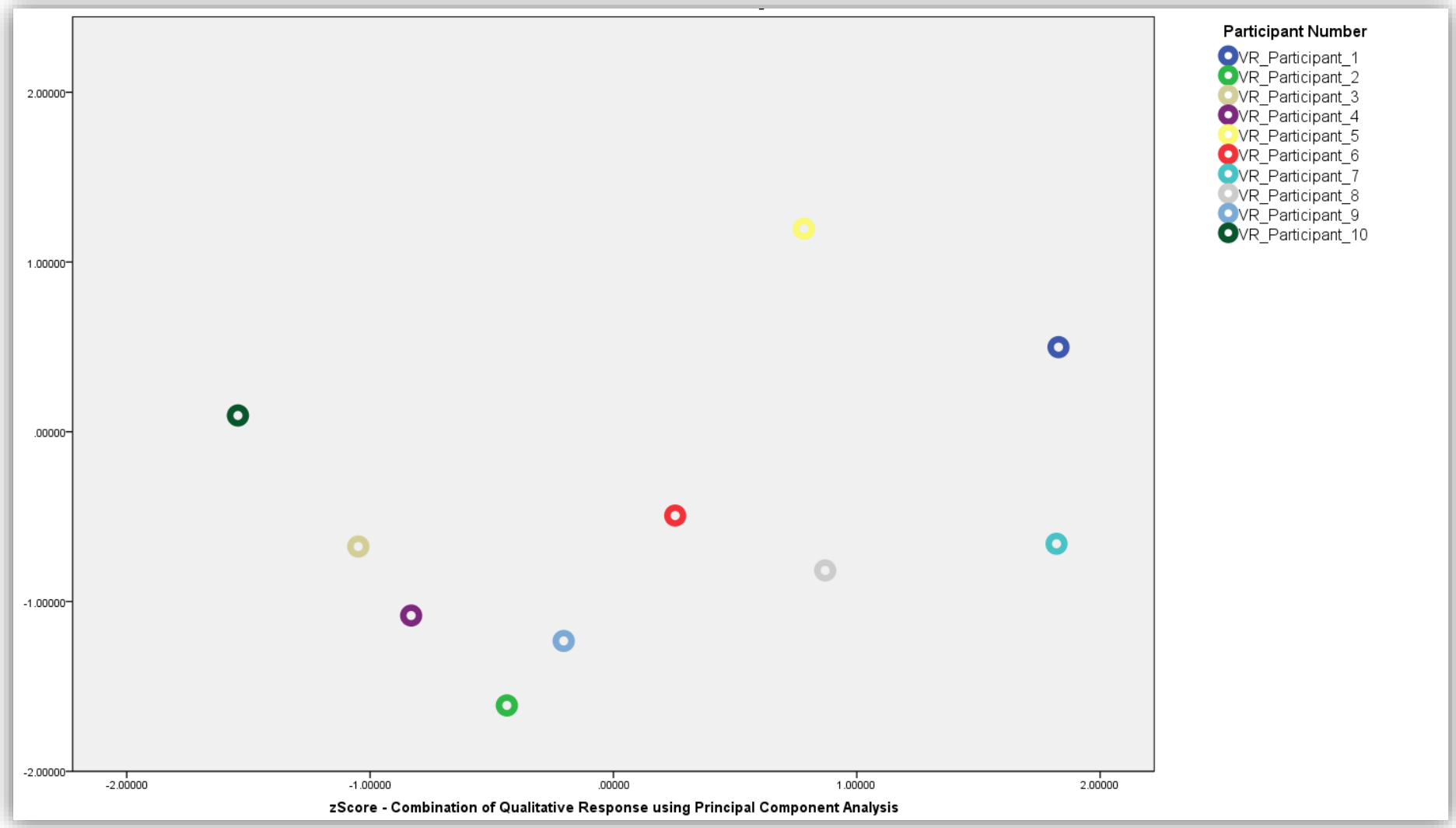


Figure 67: *Scatter plot representing the correlation between participants of the VR group that took part in the experiment.*

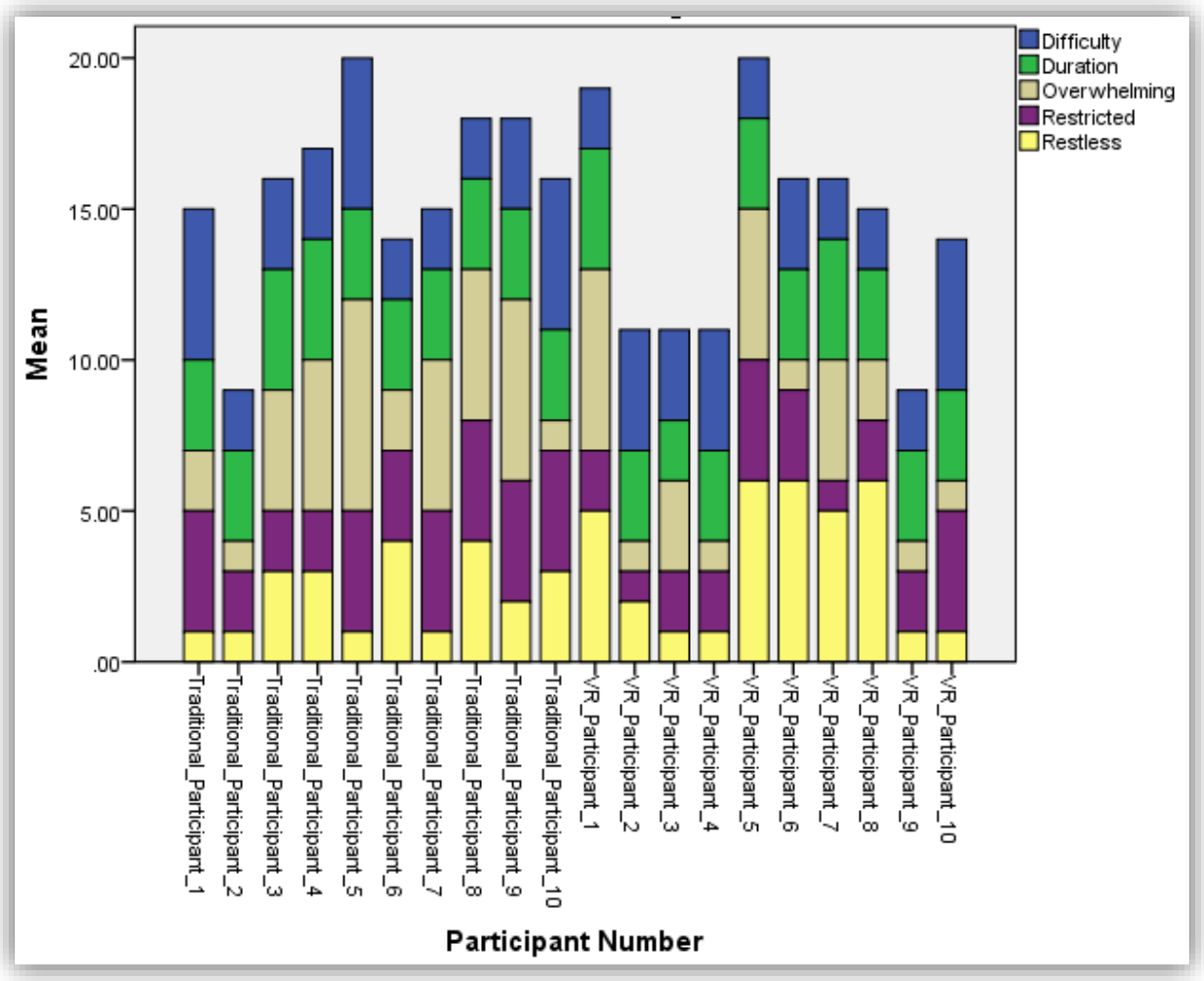


Figure 68: *Stacked bar chart showing the mean representation of the qualitative response by each participant.*

In **Figure 67**, Participants 7 and 10 seem to be quasi equidistant to each other's neighbours however both exhibit opposite qualitative response during the questionnaire. This seems to align with the fact that both participants are plotted on the opposite side of each other. The remaining participants, specifically those that are very close to each other appear to correlate in terms of qualitative response. Participants 2, 3, 4 and 9 could form two separate clusters since they have very similar attributes (see **Figure 68**) while participants 3 and 4 also share common attributes. Both participants 2 and 4

Figure 69 then shows the group of participants experiencing the TDE. The clusters exhibited in the chart seem to be more organised and systematic in comparison to that in **Figure 68**. The same applies for the isolated participants, were the further isolated participant entails higher sensory overload related symptoms. In this case, Participant 2 appears to be mostly isolated when comparing the distance to the closest neighbour. Participants 5 and 6 then follow as the next furthest participants while the remaining participants seem to be mostly clustered in separate groups.

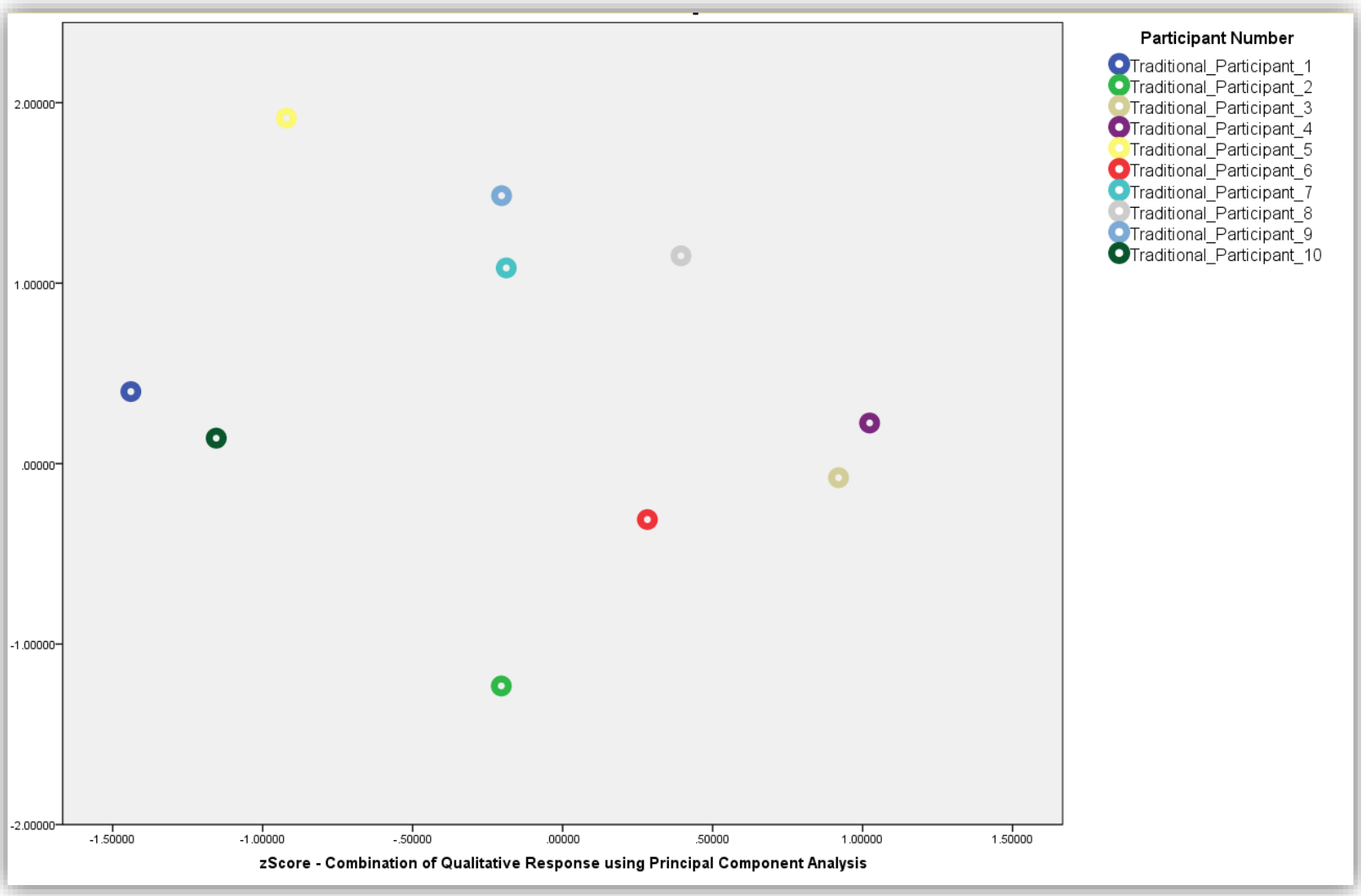


Figure 69: *Scatter plot representing the correlation between participants of the traditional group that took part in the experiment.*

8 DISCUSSION

Introduction:

The speculated hypothesis indicates that regardless of the environment setting the participant is placed in, could over stimulation of senses lead to sensory overload? Other researchers have adopted different approaches **(F. Wohlwill, 1970, p. 345)** such as the use of noisy sounds, patterned and non-patterned sounds, etc. Such studies lend to the conclusion that over stimulation of senses over a brief period of time may impact the participants' state or performance.

Both the heart rate and galvanic skin response data gathered from participants seemed sufficient during the experiments. Upon further analysis, it seemed that I had not saved copies of raw data for the VR group of participants at run time as I had done for the Traditional group of participants. This restricted my discussion further as it was expected to discuss the comparisons both quantitatively and qualitatively. As such, several correlations through GSR levels were found between both groups of participants and thus were described to aid any conclusions.

Discussion:

In all, there were a total of 16 males and 4 females that took part in the experiments. The participants varied in marital status and were intentionally selected to do so except for the last few which were relatively hard to find. Participants ranged from being in a relationship, marriage with kids and single.

This allowed me to sample different backgrounds and notice any significant difference between them. It is worth noting that one participant was colour blind and chosen specifically to test whether it impacts the load on his senses. This however did not appear to make any difference as this participant was accustomed to such a condition and had adapted to this difference.

Hourly use of computers varied from less than one hour to greater than 13 hours per day. The mix appeared to be biased towards the larger number of hours. In general, the group of VR participants were generally knowledgeable or already have had an experience with a HMD prior to this experiment. Overall three participants from this group had not be previously exposed to it and effectively impacted their performance more than those who had already experienced it before.

Although a number of participants acknowledged the fact they knew or had a vague idea of what sensory overload entails, few actually believed they exhibit such symptoms on a regular basis. The reader may view the HR and GSR line charts in **Appendix: Section 2 – Line charts for Traditional participants**. These charts depict the overall levels exhibited by the Traditional group of participants. It appeared that there were similar peak intensities for each participant at similar time periods. Participants realised the music build-up affecting their visualisation as time went by which seemed to correlate with the aspect of peak intensities. As explained in the introduction of this section, it was not possible to sufficiently display levels of the VR group of participants.

Prior to the start of the visualisation, the majority of participants reported to not feeling tired. Six participants reported feeling tired while the remaining four participants provided additional comments when yes/no did not apply for them. The participants were then given a quiz (cognitive test) on country flags which was used to measure the response time taken to complete. Based on the results obtained, most participants did not seem to be largely affected by the experiment however the majority of participants found it to be on par or easier to complete in terms of difficulty. It is worth noting however, that several participants took longer to decide their final choice in the test. This was interesting to note as it appeared that over stimulation of senses might have psychologically impacted their thought process.

Several participants reported feeling a sense of restlessness, however the majority appeared to strongly disagree. It is worth pointing out that a verbal report was taken for each participant by discussing their overall experience for a few minutes. Restlessness varied from one group to another since both had different factors to consider. In the case of HMD, considering the participants held up the HMD by hand, many appeared to get tired quite fast and unfortunately the screen panned either to the left or right at certain periods. It was expected initially expected that HMD participants would be impacted more than those in the Traditional group. Contrary to this belief, Traditional participants appeared to reveal more sensory overload related symptoms at moments, far more apparent than the other group. This could lend to the fact that the standalone version of the visualisation required mouse interaction in order to destroy the boxes on screen. However the user was restricted by 60° view such that no rotation was allowed. Therefore the participant could actively keep up with the boxes approaching the viewer from the side. This aspect may have impacted the level of anticipation in which participants could prevent boxes from getting too near, thus in effect cannot be controlled by the participant.

Given the visualisations in **Section 7: Results**, there seemed to no relation between participants in both groups feeling restless. Though when looking at the statistics produced by Principal Component Analysis, the reproduced correlation matrix for the qualitative results seem to indicate some interesting results as seen in **Figure 51**. Although it is worth mentioning that due to the small number of participants measured, there exists an equal distribution for those who did not feel restless and those that either felt different or actually started to feel restless. Having more participants would certainly help the research process however due to time restriction for both the project and studies, 20 participants were sufficient to determine some indicative results. This was deemed so as each experiment duration varied from 20 to 60 minutes which in aggregate was very time consuming.

Although most participants found the fragrances to be pleasant, this must be factored into the equation when considering those participants which exhibited sensory overload related symptoms. By looking closely at the line charts for participant 4 and 5 in **Appendix: Section 2 – Line charts for Traditional participants**, the peaks of both HR and GSR seem to match with the period when air freshener was used four times throughout each experiment. Thus both participants appear to exhibit similar patterns at the moment more stimulus (i.e. air freshener) was added into the environment and the participant started to experience it.

As such data varied from one participant to another, which may have been due to external variables or conditions, and also from the lack of cooperation by participants' that moved the sensors too much. It is also worth mentioning that due to the change of locations, the hardware had to be re-assembled every new location which potentially could have altered some of the data obtained however this is not guaranteed. Further, this seemed to align with over stimulation of senses. Regardless of what the participant thought they felt, their inner state dictated by the HR and GSR seemed to provide close matches between one another. This further adds on to the notion that over stimulation may still impact participants regardless of whether or not they perceived it. Thus this close match in data lends weight to the need for Intelligent Agents in cases such as work environments, accident prone environments and other cases where maximum focus is required.

On the whole regardless of the environment, the majority of participants appeared to find it increasingly difficult to cope with the visualisation over time. Many participants verbally reported a close resemblance between their agitated state and the level of speed at which the boxes within the visualisation was approaching them. Overall however not much attention was paid to the pulse rate of the boxes however it was denoted by two participants which are VR participant 8 and Traditional participant 5. This was further analysed in Figure 51 in **Section 7: Results** where VR participant 8 is denoted as point 8 in the chart while Traditional participant 5 is represented as point 15. Both participants appear to be isolated, though point 15 seems to be more isolated. This in fact makes a lot of sense considering the fact that both participants exhibited some levels of sensory overload related symptoms.

Participants appeared to vary in opinion as to what they considered overwhelming. In general the majority of participants, regardless of the medium used, did not appear to find it the visualisation overwhelming. That being said, those participants that found it slightly-to-highly overwhelming coincide with those participants that are isolated in **Figures 51**, **52** and **54**. It could be said that those participants in the chart that do not appear to form part of a cluster reveal that they are most likely the participants exhibiting sensory overload related symptoms. In fact those participants that are isolated which also found the visualisation to be overwhelming reported feeling anxious, irritated and contemplated terminating the experiment at their own discretion. Although not one participant terminated the experiment, it was very interesting to find out that several participants actually considered it while a couple of the traditional participants (specifically participant 4 and 5) were on the verge of termination, as discussed post-experiment stage.

Interestingly enough, although several participants contemplated the termination of experiment, not one participant found the experiment to be of very long duration. This seemed to contradict the fact that they would have terminated. This unfortunately has no concrete conclusion as to why they did not find it to be long if they would have preferred terminating the experiment, other the fact that the visualisation was overwhelming which resulted in those participants exhibiting sensory overload related symptoms.

Contrary to initial expectations, few participants actually felt more tired after the visualisation while the majority felt more alert. This could mean two things, either they

felt more alert however less focused or they are were alert and more focused. Either of these two could be further tested and analysed however this was beyond my scope of research with the emphasis being placed on the aspects of sensory overload instead of focus. Those participants which actually felt more tired after the experiment related mostly to the group of VR participants, since it involved more physical activity than the group of Traditional participants.

When looking at the verbal response with regards to the visualisation, the participants reported a variety of relations between what they were feeling and what was shown on screen. As previously described, several participants indicated the relation of the boxes speed as they approached the participant. Many described this as to closely relating to how fast they were destroying boxes, the faster the boxes approached them. This was a remarkable result since it goes to show how stable the GSR measurement actually was and showing little real-time delay when responding to the visualisation. Overall the majority of participants reported a relation between both the pulse and speed properties of the visualisation to their physiological state i.e. HR and GSR. Those participants that found no relation between their state and the visualisation reported not paying any attention to such details as their focus was mainly on destruction or simply spectating. It is worth noting that only two participants did not actively interact with the visualisation. These happened to be participant 5 and participant 10 of the Traditional group.

Interestingly enough, participant 5 happened to exhibit some sensory overload related symptoms and reported the inclination to start interacting with the visualisation however concluded that he did not feel the need to do so. This was peculiar to observe as the participant reported being irritated and anxious when the screen was overpopulated with boxes and calmed down when the screen cleared slightly. Upon discussion as to why the participant had not interacted with the visualisation, even though they found it to be somewhat overwhelming, the participant response was that of no interest to interact. The participant simply wanted to observe what would happen over time, even though he wasn't advised to do so.

Participants reported both the music and visualisation impacted their performance. Those participants who appreciated every song were not impacted by music however the visualisation appeared to overwhelm them to a certain extent. On the other hand, those participants which happened to dislike either one or more songs seemed to become more restless and agitated as time went by. In fact, one participant verbally reported back with a headache once complete while another participant was very close to closing the application mid-experiment, as explained in the previous paragraph. This clearly indicated that such participants were suffering from sensory overload symptoms when comparing them to the list of related symptoms. Although these participants were unaware of such symptoms, upon further discussion and realisation what sensory overload entailed, both participants admitted to feelings along those lines. In this regard, I consider these two participants to be successful candidates for over stimulation of senses within my project.

The reproduced correlation chart in **Figure 45** represents the correlation between participants' recorded GSR data. Although this simply indicates how one participant compares to another in terms of GSR, it is still useful to correlate participants using such

data. This is due to the fact that once the participant started to experience new stimulus, the charts depict an elevation of GSR levels. Therefore, although the GSR readings varies from one person to another, it still is helpful to look at how each participant elevated in levels from their benchmark sample as described in the experiment introduction in the results section.

To further visually represent the values shown in **Figure 45**, **Figures 46** through **50** represent the five types of correlation that could be extracted from the chart. As described in the results section, those who had a high-positive correlation consisted of data points which were clustered together. Similarly, those participants with a low-positive correlation also revealed clusters of data points, however in this case they were more spread out as they approached the no-correlation zone. Those participants which had no correlation with another participant appeared to consist of data points that were simply spread out along the chart with no relation to each other. Negative correlations still indicate a correlation, however in this case, an inverse correlation. This essentially entails that as one participant felt more stressed, the other participant appeared to feel less stressed which could be seen in **Figure 48**. Thus, those participants with high-negative correlations also appeared with clustered data points however on the opposite side of the chart (see **Figures 49** and **50**).

As explained a few paragraphs back, **Figure 51** represents the Principal Component Analysis of all participants' qualitative data with regards to how each correlate with each other. Every data point in the chart represents a participant. Every participant consists of five attributes which define how they correlate to each other. This could be seen in **Figure 54** which shows a stacked bar chart depicting the mean of the qualitative response gathered from participants throughout the questionnaire. **Figure 53** represents the how each of these attributes correlate with the other. In this case, the attributes *Restless* and *Restriction* appeared to be the highest-positive correlation, followed by *Overwhelming* and *Restless*. These comparisons further add to the speculation that these three attributes where mainly related to the symptoms of sensory overload, as experienced by participants.

It is worth pointing out two biased participants within both groups. Participant 7 of the VR group had taken part in previous prototype tests and thus had a high-knowledgeable bias for the experiment while Participant 8 of the Traditional group was also another participant that took part in prototype tests. By further analysing **Figures 53** and **55**, both these participants appear to be slightly isolated however still within a respectable distance to another participant. This entails that regardless of the bias, being in separate groups, academic history, age and background, both participants still seem to exhibit minor sensory overload related symptoms which further strengthens the speculation as described in the introduction of this section.

9 LIMITATIONS

9.1 REQUESTING URLS

During the development of the output module, I came across a strong limitation in Unity3D which prevented me from solving the issue for this project. The problem relates to the moment the application retrieves a JSON file from the server.

Unity3D's free-license mode restricts the user to using threads with a hands-on approach. Instead they offer a coroutine function which essentially functions in the same manner as a traditional thread in java with the exception that the engine handles the event instead of the user. An applicable scenario for coroutines is to process data in the background without blocking the Update() thread.

The coroutine creates an event and refers the process to the engine to manage it while it processes data. Once the process is complete, the coroutine merges back with the Update() thread and continues to operate. In my case, the issue occurs once I have successfully downloaded the JSON file from the server.

Scenario:

I create a coroutine event with a request to download the file and gets referred to the engine. The Update() thread maintains operation and processing of the visualisation while the request is being processed. Once the coroutine merges back with the Update() thread, the GUI blocks for a very short period of time. Upon further investigation, I discovered that it was the moment the file is received from the server.

Having attempting to solve this issue for a few days, I decided to stop my attempt as every solution I encountered did not work out well. The only solution that exists at this point in time is to purchase the Professional license which was completely financially infeasible.

9.2 MITIGATION:

In attempt to mitigate this issue rather than solve it, I had implemented a minor-fix which set a 10 second time-window in between requests. This allowed the user to sufficiently interact with the visualisation with minimal distraction.

Overall this issue is considered perhaps one of the most important problems that must be dealt with in the future. That said, this problem turned out to be useful at times for participants as it acted as an interrupter during the experiment which could have irritated the participants. It is worth noting however that only one participant pointed out the interrupt problem.

9.3 DATA PROCESSING MODULE OCCASIONALLY LOST PACKETS IN TRANSIT

Another issue that was not solvable in due time related to the Data Processing Module instead. The severity of this issue is rather high considering loss of data. While testing the data transmission it appeared that as time went by, packets started to get lost mid-

transmission. One of the options considered is the FTP failing however cannot conclude what the issue pertains to.

Throughout each experiment, participants transmit some 70 to 90 JSON files to the server from the Data Processing Module and in turn, the Output Module only extracts roughly 40 to 50 of them. This is obviously not desirable however is unavoidable at this point in time.

Given time restrictions and given a wealth of time in attempt to solve or mitigate the issue, I opted to go along with this issue. My suspicions involve the thread handling in Java since a new thread is created to handle each file being sent to the server. The process of uploading to the server involves a series of events that undergo data retrieval, fuzzification and JSON object wrapping. Once these events are complete, the JSON object is then sent as a JSON file to the server.

Considering this process and the requirement to unravel the JSON object in the Output module lends to the idea that the lengthy process results in packets being lost/not sent and less packets being extracted in the Output module. Although considered a limitation, the number of lost packets don't outweigh the number of received packets. Therefore it was acceptable to lose some packets while still obtaining relatively accurate measures.

9.4 DEVICES

Given the restricted circumstances of completing the project in good time and providing an academically worthy dissertation, I opted to use devices that are within reach rather than higher risk (in terms of delivery and availability), less familiar technologies.

There are multitudes of vendors that sell a variety of HR and GSR sensors though I opted to seek out those that communicated with an Arduino micro-controller given my previous experience with an Arduino. Some other types of devices such as wearable devices are able to gauge relatively accurate physiological data however I preferred to obtain electrical signals to work on byte level.

10 CONCLUSION

Considering both the qualitative and quantitative data gathered from participants, the results indicate a relation between over stimulation of senses in both traditional and a HMD environment and sensory overload related symptoms. The restricted number of participants' makes it difficult to completely determine such results though it would serve well to investigate well over 50 participants to obtain more accurate results.

Though not conclusive, the results proved to be positive as the hypothesis set out to be quite interesting to investigate and only encourages me to take it one step further. Given the trend of HCI principles being applied in modern day technology, it would be of great advantage to involve intelligent agents that are able to capture physiological data from individuals in order to adapt our interfaces for a more comforting, productive and safe environment.

As described in **Section 10.1. Future Work** below, there are several applicable areas that could utilise this research. Such applications involve accident prone environments, moving vehicles, interaction design, home and office environments and many more.

10.1 FUTURE WORK

Several technologies such as Fuzzy Logic, HMDs, HR and GSR sensors were examined in this project. It makes sense for other authors' to have used and will use different means to measure individuals' response with respect to research related to this project. An interesting and slightly expected correlation that resulted from this project was the impact of over stimulation on participants' psychophysiological state regardless of the environment setting (HMD vs. TDE).

This project motivated me to further my research on sensory overload within the context of Interaction Design as I can see many situations where this research is applicable in the industry. Typical applications involve accident prone environments, moving vehicles, home and work environments.

In the case of accident prone environments, this research could be applied in the case of heavy machinery. Due to the dangerous nature of the job, complete focus and attention is required to complete the task. Thus by integrating a set of sensors which continuously detect the workers physiological state, they could be used to measure how distracted, attentive or focused they are as well as helping them conduct their operations.

In moving vehicles such as typical cars, sensors could be employed to measure their physiological state, similar to the case of accident prone environments. This way, the car would be intelligent enough to steer the owner in the safest possible manner. This may involve a non-intrusive agent which could set a maximum threshold of speed which could be vary depending on the individual state. If the individual is 90% attentive then it sets the maximum speed to 90% of the maximum speed limit, for example.

Conversely, if the individual happens to be less than 50% attentive, the car would alarm the individual to put the car aside and would not allow the driver to start the car until it considers the individual to be fit to drive. This notion could be applied in cases such as

drink-driving. Similar technology could be integrated to measure the level of alcohol that the individual has consumed in a non-intrusive manner. Thus it could prevent the car from starting should the car indicate that the individual is unfit to drive.

Another application for this research could involve an agent that measures the individuals' physiological state to adapt the User Interface that they are interacting with. This way the individual could interact with an interface which is reduced in clutter in order to maintain the users' focus while working. This is beneficial for such environments as the over stimulation of senses can lead to less-productive work, as described in **Section 5: Literature Review – 5.4.2. Sensory Overload**.

The list of similar applications is countless and thus I actively encourage the use of this research to further any academic curiosity one may have. I hope this work could be of help to the academic community.

11 EVALUATION:

Overall, I would consider this project to be largely successful. Both in terms of successful implementation and execution as well as the results obtained. As such, these properties seem to align with the predefined aims and objectives, stated in the introduction.

Through the analysed and discussed results, I learned that Traditional participants appeared tend to be more affected by over stimulation of senses rather than the VR participants, contrary to what was initially expected prior to the experimentation stage. It was largely expected that due to being immersed within a virtual environment and being "closer" to the screen would entail such participants' being more impacted than those in Traditional. In fact I initially assumed that those participants taking part in the Traditional experiments would find themselves bored and reluctant to resume. It appeared that the immersion factor contributed to participants feeling more alert and awake than before however some participants were more tired than before due to the wire restrictions and holding the VR display.

As I was proof reading the project, I came across an interesting study by (Harrison & Loui, 2016) which went into the merits of how music leads to psychophysiological experiences in music. This was an astounding find since it served as the basis for the inspiration behind my thesis. It's a shame that I encountered this research so late in the project considering that this was published back in 2014. Upon quick inspection of the paper, the authors looked into the elicitation of intense emotional and psychophysiological response that individuals' exhibit when listening to music of their liking.

This study seemed to perfectly align with what I was hoping to elicit from participants initially. Given participants' may or may not appreciate the music selection, those who appreciate it might exhibit positive psychophysiological response while those who do not might exhibit a stronger sense of anxiety, irritation or distraction due to the fact that such individual might not like the music. In the end, both could relate to each other in terms of strong response however one might exhibit a positive response while the other exhibits a negative one.

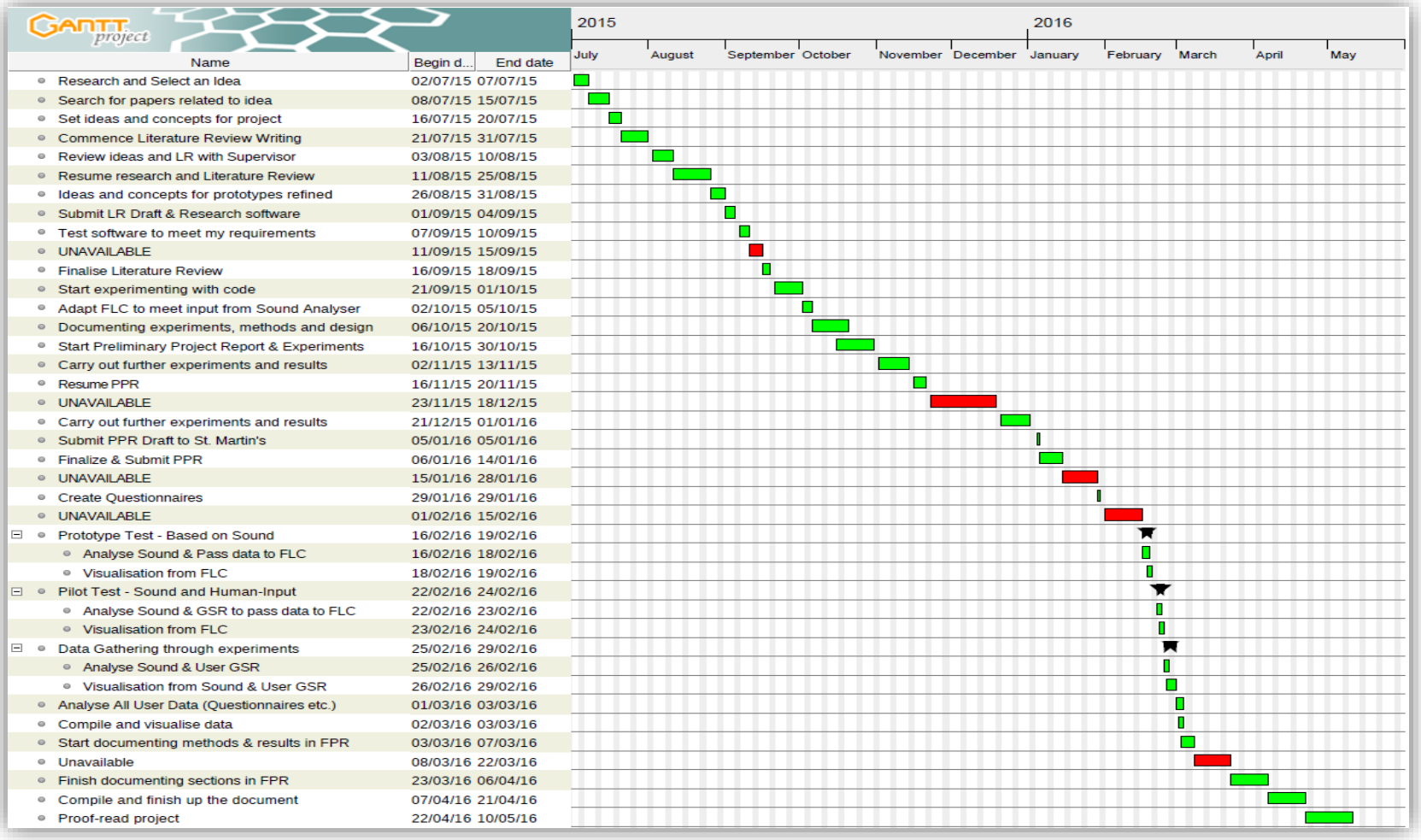
Overall given the results analysed and discussed in Sections 7 and 8 as well as the limitations in Section 9, I believe that I have successfully met the pre-defined aims and objectives however slightly impacted by the methodological process.

What would I do differently?

One of the main things I would consider a different approach pertains to the methodological approach of this project. Although it was sufficient to meet the pre-defined aims and objectives, the limitations addressed in Section 9 may have impacted the final results obtained.

12 APPENDIX

Project plan



Name	Begin d...	End date
Research and Select an Idea	02/07/15	07/07/15
Search for papers related to idea	08/07/15	15/07/15
Set ideas and concepts for project	16/07/15	20/07/15
Commence Literature Review Writing	21/07/15	31/07/15
Review ideas and LR with Supervisor	03/08/15	10/08/15
Resume research and Literature Review	11/08/15	25/08/15
Ideas and concepts for prototypes refined	26/08/15	31/08/15
Submit LR Draft & Research software	01/09/15	04/09/15
Test software to meet my requirements	07/09/15	10/09/15
UNAVAILABLE	11/09/15	15/09/15
Finalise Literature Review	16/09/15	18/09/15
Start experimenting with code	21/09/15	01/10/15
Adapt FLC to meet input from Sound Analyser	02/10/15	05/10/15
Documenting experiments, methods and design	06/10/15	20/10/15
Start Preliminary Project Report & Experiments	16/10/15	30/10/15
Carry out further experiments and results	02/11/15	13/11/15
Resume PPR	16/11/15	20/11/15
UNAVAILABLE	23/11/15	18/12/15
Carry out further experiments and results	21/12/15	01/01/16
Submit PPR Draft to St. Martin's	05/01/16	05/01/16
Finalize & Submit PPR	06/01/16	14/01/16
UNAVAILABLE	15/01/16	28/01/16
Create Questionnaires	29/01/16	29/01/16
UNAVAILABLE	01/02/16	15/02/16
Prototype Test - Based on Sound	16/02/16	19/02/16
Analyse Sound & Pass data to FLC	16/02/16	18/02/16
Visualisation from FLC	18/02/16	19/02/16
Pilot Test - Sound and Human-Input	22/02/16	24/02/16
Analyse Sound & GSR to pass data to FLC	22/02/16	23/02/16
Visualisation from FLC	23/02/16	24/02/16
Data Gathering through experiments	25/02/16	29/02/16
Analyse Sound & User GSR	25/02/16	26/02/16
Visualisation from Sound & User GSR	26/02/16	29/02/16
Analyse All User Data (Questionnaires etc.)	01/03/16	03/03/16
Compile and visualise data	02/03/16	03/03/16
Start documenting methods & results in FPR	03/03/16	07/03/16
Unavailable	08/03/16	22/03/16
Finish documenting sections in FPR	23/03/16	06/04/16
Compile and finish up the document	07/04/16	21/04/16
Proof-read project	22/04/16	10/05/16

12.1 SECTION 1 - METHODS:

12.1.1 Hardware Configurations

List all the hardware involved

- Arduino
- Heart Rate Sensor
- Galvanic Skin Response Sensor
- Google Cardboard Headset
- Headphones
- Laptop

A variety of hardware were required for the purpose of this project. Considering the research strategy and literature review, it was deduced that the Galvanic Skin Response and Heart Rate sensors were required. Since the sensors were purchased from Seeed Studio (Seeed, 2015) through Amazon, I was able to guarantee a fall-back should anything go wrong with the sensors.

These sensors are built to work with their own base module or other modules such as the Arduino controller. The Arduino language is merely a set of C/C++ functions that can be used in the sketches **(Arduino, 2015)**.

When integrating Processing IDE (PApplet) in Eclipse, the working module must extend the PApplet class so as to run the setup() and draw() methods provided by the PApplet. The issue is that Processing implements its own Serial library which is deprecated when attempting to use it within Eclipse. This was solved by importing the RXTX and JSSC libraries which are the modern-day Serial communication libraries which is implemented using Java programming language.

Finally, the output module uses Unity3D engine which allowed for communication with the FTP to download the files containing the fuzzified data that was stored in the Input module. All of this can be implemented in C# or JavaScript however for simplicity sake and given previous experience, all of it was developed using C# programming language.

12.1.2 **Circuit Diagram for Arduino, Breadboard and Sensors**

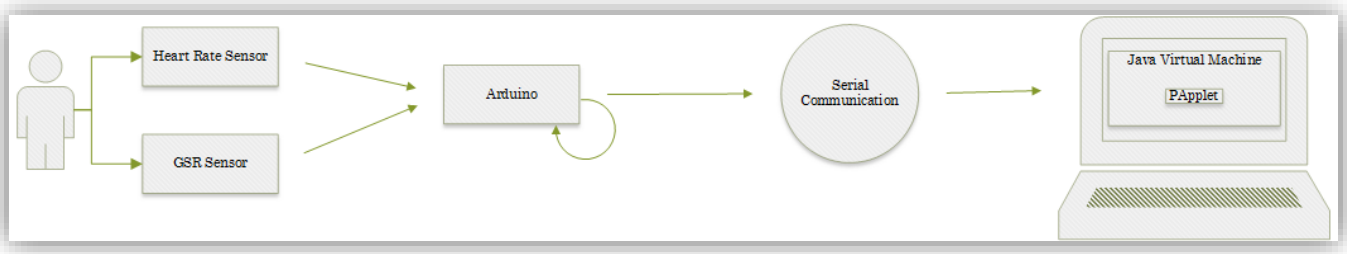


Figure 70: *From human input to serial communication with a computer and the Java Environment with PApplet integrated within.*

12.1.3 **Arduino Program Code**

List the program code used for the Arduino which allowed me to receive data from the Arduino fed by the sensory human input.

```
// GSR sensor variables
const int GSR = A3;  // select the input pin for the GSR
const int numReadings = 10;
// Readings for averaging of input
int readings[numReadings];              // the readings from the analog input
int readIndex = 0;                      // the index of the current reading
int total = 0;                          // the running total
int average = 0;                        // the average
int gsrValue = 0;

// GSR Time variables
unsigned long time;
int secForGSR;
int curMillisForGSR;
int preMillisForGSR;

// HR sensor variables
#define Heart 2                         //Attach the Grove Ear-clip sensor to digital pin 2.
int val = 0;
boolean beat = false;                   /* This "beat" variable is used to control the timing of the Serial
communication
                                        so that data is only sent when there is a "change" in digital readings. */
unsigned long interval = 25;            // The time needed to wait
unsigned long previousMillis = 0;       // millis() returns an unsigned long

void setup() {
  long sum = 0;
  Serial.begin(9600);                   //Initialise serial communication
  pinMode(Heart, INPUT);                //Set digital pin 2 (heart rate sensor pin) as an INPUT
  secForGSR = 1;                        // How often do we get a GSR reading
  curMillisForGSR = 0;
  preMillisForGSR = -1;

  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings[thisReading] = 0;
  }
}

void loop() {
```

```
// Galvanic Skin Response Sensor Code
time = millis();


curMillisForGSR = time / (secForGSR * 1000);
if(curMillisForGSR != preMillisForGSR) {
  // Read GSR sensor and send over Serial port
  total = total - readings[readIndex];
  gsrValue = analogRead(GSR);
  if (gsrValue > 0) {
    readings[readIndex] = gsrValue;
    total = total + readings[readIndex];
    readIndex += 1;

    if (readIndex >= numReadings) {
      readIndex = 0;
    }
    average = total/numReadings;
    Serial.print("g");
    Serial.println(average);
  }

  preMillisForGSR = curMillisForGSR;
}

unsigned long currentMillisForHR = millis();   // Grab current time for HR
if ((unsigned long) (currentMillisForHR - previousMillis) >= interval) {
  // Heart Rate Sensor Code
  val = digitalRead(Heart);
  if(val>0){              //The heart rate sensor will trigger HIGH when there is a heart beat
    if(!beat){                        //Only send data when it first discovers a heart beat - otherwise it
will send a high value multiple times
      beat=true;                  //By changing the beat variable to true, it stops further transmissions
of the high signal
      Serial.print("h");
      Serial.println(1023);           //Send the high value to the computer via Serial communication.
    }
   } else {                          //If the reading is LOW,
    if(beat){                         //and if this has just changed from HIGH to LOW (first low reading)
      beat=false;                     //change the beat variable to false (to stop multiple transmissions)
      Serial.print("h");
      Serial.println(0);              //then send a low value to the computer via Serial communication.
    }
   }
   previousMillis = millis();
  }
}
```

12.1.4 **Data Processing Module**

12.1.4.1 Classifying Serial Data

This is an inner class to the ControllerInputExtractor.java which can be seen in the complete package within the uploaded archive.

```
private class PortReader implements SerialPortEventListener {
    ArrayList<String> hrList = new ArrayList<String>();
    ArrayList<String> gsrList = new ArrayList<String>();
    ArrayList<String> finalHRList = new ArrayList<String>();
    ArrayList<String> finalGSRList = new ArrayList<String>();
    String GSRpreviousString = "", HRpreviousString = "", previousString = "";
    int j = 0, k = 0;

    //    http://www.codeproject.com/Tips/801262/Sending-and-receiving-strings-from-COM-port-
via-jS
    @Override
    public void serialEvent(SerialPortEvent arg0) {
      if (arg0.isRXCHAR() && arg0.getEventValue() > 0) {
       try {
          comPortString = myPort.readString(arg0.getEventValue());

          // Read this data into the comPortString variable.
          String trimmedComPortString = comPortString.trim();

          // If the comPortString variable is not NULL then
           if(trimmedComPortString != null) {
            if (trimmedComPortString.substring(0, 1).equals("h")) {
              hrList.add(trimmedComPortString);
              HRpreviousString = trimmedComPortString;
              previousString = HRpreviousString;
              k += 1;
            } else if (trimmedComPortString.substring(0,  1).equals("g")) {
              gsrList.add(trimmedComPortString);
              GSRpreviousString = trimmedComPortString;
              previousString = GSRpreviousString;
              j += 1;

            } else if (Character.isDigit(trimmedComPortString.charAt(0))) {
              if (previousString.charAt(0) == 'g') {
                // append to the last string in the gsr list
                String newstring = gsrList.get(gsrList.size()-1);
                StringBuilder finalString = new StringBuilder();
                finalString.append(newstring);
                finalString.append(trimmedComPortString);
                finalGSRList.add(finalString.toString());
                dataContainerClass.setGsrList(finalGSRList);

                // Get the data value
                int i = Integer.parseInt(finalString.toString().substring(1));

                // Check if we detected a GSR signal
                if (!gsrBeat) {
                  // Sample the GSR and add to list for averaging
```

```
                    triggerGSR(millis(), i);
                    // Indicate that GSR signal detected
                    gsrBeat = true;
                  }

                 // If the value is less than the threshold, allow to re-read
                 if (i < 389) {
                  gsrBeat = false;
                 }

                } else if (previousString.charAt(0) == 'h') {
                  // append to the last string in the hr list
                  String newstring = hrList.get(hrList.size()-1);
                  StringBuilder finalString = new StringBuilder();
                  finalString.append(newstring);
                  finalString.append(trimmedComPortString);
                  finalHRList.add(finalString.toString());
                  dataContainerClass.setHrList(finalHRList);
                  // Get the data value
                  int i = (int)(Integer.parseInt(finalString.toString().substring(1)));

                  // Convert the integer into a float value.
                 arduinoValue = (float)i;
                 if (!beat){
                  // When a beat is detected, the "trigger" method is called.
                  if(arduinoValue>0.0) {
                    // millis() creates a timeStamp of when the beat occured.
                    triggerHR(millis());

                    // The beat variable is changed to TRUE to register that a beat has been
detected.
                    beat=true;
                  }
                 }
                 // When the Arduino value returns back to zero,
                 // we will need to change the beat status to FALSE.
                 if (arduinoValue<1.0){
                   beat = false;
                 }
               }
             }
            }
         } catch(Exception e) {
           e.printStackTrace();
         }
        }
     }
   }
```

12.1.4.2 Mapping the physical to the virtual world

```
// Block definition (there may be more than one block per file)
FUNCTION_BLOCK flcserver

// Define input variables
VAR_INPUT
   frequency : REAL;
   amplitude : REAL;
   heartrate : REAL;
   galvanic :  REAL;
END_VAR

// Define output variable
VAR_OUTPUT
   colour : REAL;
   size: REAL;
   pulse: REAL;
   speed: REAL;
END_VAR

// Fuzzify input variable 'frequency'
//FUZZIFY frequency
//  TERM low := (0, 0) (5512.5, 1) (11025, 0);
//    TERM medium := (5512.5, 0) (11025,1) (16537.5,0);
//    TERM high := (11025, 0) (16537.5, 1) (22050, 0);
//END_FUZZIFY
FUZZIFY frequency
   TERM subbass := (0, 1) (50, 1) (100, 0);
   TERM midbass := (80, 0) (290, 1) (500, 0);
   TERM midrange := (400, 0) (1200, 1) (2000, 0);
   TERM uppermid := (1000, 0) (3500,1) (6000,0);
   TERM highfreq := (4000, 0) (8000,1) (12000,0);
   TERM veryhighfreq := (10000, 0) (15000, 1) (22000, 1);
END_FUZZIFY

// Fuzzify input variable 'amplitude'
FUZZIFY amplitude
   TERM very_low := (0, 0.75) (0.5, 1) (6, 0);
   TERM low := (1,0) (5.5,1) (10,0);
   TERM normal := (4,0) (14.5,1) (25,0);
   TERM high := (7,0) (26.5,1) (55,0);
   TERM very_high := (30,0) (55,1) (80,0);
END_FUZZIFY


// Fuzzify input variable 'heartrate'
FUZZIFY heartrate
// Source: http://www.weightwatchers.com/util/art/index_art.aspx?tabnum=1&art_id=20971
   TERM low := (0, 0) (68, 1.0) (93.1, 1) (93.1, 0); //(76, 1.0)
   TERM steady := (92.9, 0) (92.9,1.0) (119.1, 1.0) (119.1, 0); //(105.5,1)
   TERM high := (118.9,0) (118.9, 1.0) (170, 1.0);
END_FUZZIFY

// Fuzzify input variable 'galvanic'
FUZZIFY galvanic
   TERM relaxed := (0, 0.75) (38.5, 1) (77, 0);
   TERM focused := (70,0) (151.5,1) (233,0);
```

```
    TERM stable := (230,0) (251,1) (272,0);
    TERM nervous := (270,0) (310,1) (350,0);
    TERM stressed := (311,0) (351.5,1) (392,0);

END_FUZZIFY

// Defzzzify output variable 'colour'
DEFUZZIFY colour
    TERM violet := (3800,0) (4150,1) (4500,0);
    TERM blue   := (4500,0) (4620.5,1) (4750,0);
    TERM cyan   := (4760,0) (4850.5,1) (4950,0);
    TERM green  := (4950,0) (5320.5,1) (5700,0);
    TERM yellow := (5700,0) (5800,1) (5900,0);
    TERM orange := (5900,0) (6050,1) (6200,0);
    TERM red    := (6200,0) (6850,1) (7500,0);

    // Use 'Center Of Gravity' defuzzification method
    METHOD : COG;

    // Default value is 0 (if no rule activates defuzzifier)
    DEFAULT := 0;
END_DEFUZZIFY

// Defzzzify output variable 'size'
DEFUZZIFY size
    TERM small := (0,0) (1.5,1) (3,0);
    TERM medium := (3,0) (4.5,1) (6,0);
    TERM large := (6,0) (7.5,1) (9,0);

    // Use 'Center Of Gravity' defuzzification method
    METHOD : COG;

    // Default value is 0 (if no rule activates defuzzifier)
    DEFAULT := 0;
END_DEFUZZIFY

// Defzzzify output variable 'pulse'
DEFUZZIFY pulse
    TERM slow := (0,0) (1.5,1) (3,0);
    TERM moderate := (3,0) (4.5,1) (6,0);
    TERM fast := (6,0) (7.5,1) (9,0);

    // Use 'Center Of Gravity' defuzzification method
    METHOD : COG;

    // Default value is 0 (if no rule activates defuzzifier)
    DEFAULT := 0;
END_DEFUZZIFY

// Defzzzify output variable 'speed'
DEFUZZIFY speed
    TERM veryslow := (0,0) (0.125,1) (0.25,0);
    TERM quiteslow := (0.2,0) (0.325,1) (0.45,0);
    TERM normalspeed := (0.4,0) (0.525,1) (0.65,0);
    TERM quitefast := (0.6,0) (0.725,1) (0.85,0);
    TERM veryfast := (0.8,0) (0.925,1) (1.0,0);

    // Use 'Center Of Gravity' defuzzification method
```

```
    METHOD : COG;

    // Default value is 0 (if no rule activates defuzzifier)
    DEFAULT := 0;
END_DEFUZZIFY

RULEBLOCK No1
    // Use 'min' for 'and' (also implicit use 'max'
    // for 'or' to fulfill DeMorgan's Law)
    AND : MIN;
    // Use 'min' activation method
    ACT : MIN;
    // Use 'max' accumulation method
    ACCU : MAX;

    RULE 1 : IF frequency IS uppermid
             THEN colour IS violet;

    RULE 2 : IF frequency IS highfreq
             THEN colour IS cyan;

    RULE 3 : IF frequency IS veryhighfreq
             THEN colour IS yellow;

    RULE 4 : IF amplitude IS very_low OR amplitude IS low
             THEN size IS small;

    RULE 5 : IF amplitude IS low OR amplitude IS normal
             THEN size IS medium;

    RULE 6 : IF amplitude IS normal OR amplitude IS high
             THEN size IS medium;

    RULE 7 : IF amplitude IS high OR amplitude IS very_high
             THEN size IS large;

    RULE 8 : IF heartrate IS low OR heartrate IS steady
          THEN pulse IS slow;

    RULE 9 : IF heartrate IS steady OR heartrate IS high
          THEN pulse IS moderate;

    RULE 10 : IF heartrate IS high
          THEN pulse IS fast;

END_RULEBLOCK

RULEBLOCK No2
    // Use 'min' for 'and' (also implicit use 'max'
    // for 'or' to fulfill DeMorgan's Law)
    AND : MIN;
    // Use 'min' activation method
    ACT : MIN;
    // Use 'max' accumulation method
    ACCU : MAX;

    RULE 11 : IF frequency IS uppermid OR frequency IS highfreq
          THEN colour IS blue;
```

```
    RULE 12 : IF heartrate IS low OR heartrate IS steady OR heartrate IS high
        THEN pulse IS moderate;

    RULE 13 : IF galvanic IS relaxed OR galvanic IS focused
        THEN speed IS veryfast;

    RULE 14 : IF galvanic IS focused OR galvanic IS stable
        THEN speed IS quitefast;

    RULE 15 : IF galvanic IS stable OR galvanic IS nervous
        THEN speed IS normalspeed;

    RULE 16 : IF galvanic IS nervous OR galvanic IS stressed
        THEN speed IS quiteslow;

    RULE 17 : IF galvanic IS stressed
        THEN speed IS veryslow;

    RULE 18 : IF galvanic IS stable OR galvanic IS nervous OR galvanic IS stressed
        THEN speed IS quiteslow;

    RULE 19 : IF galvanic IS relaxed OR galvanic IS focused OR galvanic IS stable
        THEN speed IS quitefast;

    RULE 20 : IF galvanic IS focused OR galvanic IS stable OR galvanic IS nervous
        THEN speed IS normalspeed;

END_RULEBLOCK

RULEBLOCK No3
    // Use 'min' for 'and' (also implicit use 'max'
    // for 'or' to fulfill DeMorgan's Law)
    AND : MIN;
    // Use 'min' activation method
    ACT : MIN;
    // Use 'max' accumulation method
    ACCU : MAX;

    RULE 21 : IF frequency IS highfreq OR frequency IS veryhighfreq
        THEN colour IS yellow;

    RULE 22 : IF frequency IS midrange OR frequency IS uppermid OR frequency IS highfreq
        THEN colour IS green;



END_RULEBLOCK
END_FUNCTION_BLOCK
```

12.1.4.3

12.1.4.4 JSON Encoder Algorithm

```
public static void main(String[] args) {
    PApplet.main(new String[] {  "com.analysers.ControllerInputExtractor" });

    String server = "www.enterdomainname.domain";
    int port = 21;
    String username = "enterUsername";
    String password = "enterPassword";

    FTPClient ftpClient = new FTPClient();
    /**
     * Wait 6 seconds before checking size of data
     */
    try {
        Thread.sleep(22000);
    } catch (InterruptedException e1) {
        System.out.println("Thread error: " + e1);
    }

    // Send data to server then logout
    try {
        Runnable helloRunnable = new Runnable() {
            public void run() {
                // If the song is still playing (i.e. the song has not ended yet)
                if (!dataContainerClass.getSongPlaying()) {
                    try {
                        // Establish connection with FTP client with server and port number
                        ftpClient.connect(server, port);
                        showServerReply(ftpClient);

                        // Check whether connection was complete
                        int replyCode = ftpClient.getReplyCode();
                        if (!FTPReply.isPositiveCompletion(replyCode)) {
                            System.out.println("Connection Failed.");
                            return;
                        }

                        // Attempt login if connection is complete and show server reply
                        boolean success = ftpClient.login(username, password);
                        showServerReply(ftpClient);

                        // Set the file type to not get corrupted messages
                        ftpClient.setFileType(FTP.BINARY_FILE_TYPE);

                        // If failed, show failed login error
                        if (!success) {
                            System.out.println("Could not login to the Server.");
                        }

                         // Change working directory to store Fuzzified Data and show server reply
                        success = ftpClient.changeWorkingDirectory("public_html/fuzzyData");
                        showServerReply(ftpClient);
                        // Display server reply if successful or failed
```

```
            if (success) {
                System.out.println("Successfully changed working directory.");
            } else {
                System.out.println("Failed to change working directory.");
            }

            // Clone extracted data to keep a copy
            dataContainerClass.cloneExtractedData();

            // Store temporary data
            ArrayList<ArrayList<String>> data = dataContainerClass.getClonedData();
            System.out.println("Data Size: " + data.size());

            // JSONObject to hold new packet to be uploaded to FTP
            JSONObject dataObject = new JSONObject();

            // Packet number
            int pivot = 0;

            // For every piece of data acquired from the user
            for (int j = 0; j < data.size(); j++) {
                // Get the set of data acquired from Analysis
                ArrayList<String> listOfData = data.get(j);

                // Perform Fuzzy Logic Inference on the acquired Data
                dataObject = runFuzzyLogic(pivot, listOfData, data);
            }

            // Create a null instance due to Fuzzy System not finding the instance
            FileWriter localFile = null;
            try {
                System.out.println("Containing Objects: " + dataObject.toString());

                // Create a localFile with the latest Data Object
                localFile = new
FileWriter("C:/Users/James/Documents/THESIS/Experiments/JSON_Objects/" + data.size() +
"_fuzzySet.json");

                // Write the JSONObject to file
                localFile.write(dataObject.toString());
                localFile.flush();
                localFile.close();

                // Create a remote file name to store on the server
                String remoteFile = data.size() + "_fuzzySet.json";

                // Send the JSON data to the stream
                InputStream inputStream = new
FileInputStream("C:/Users/James/Documents/THESIS/Experiments/JSON_Objects/" +
data.size() + "_fuzzySet.json");
                System.out.println("Starting to upload the local file...");

                // Upload the remote file to the server
```

```
                    boolean done = ftpClient.storeFile(remoteFile, inputStream);
                    inputStream.close();

                    // If successfull upload - confirm upload
                    if (done) {
                        System.out.println("We have just uploaded the " + data.size() + " file");
                    } else {
                        System.out.println("Uploading to FTP Server has failed.");
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }
            } catch(IOException e) {
                System.out.println("Something went wrong in the process!");
                e.printStackTrace();
            } finally {
                try {
                    // Disconnect from server safely
                    if (ftpClient.isConnected()) {
                        System.out.println("We are about to disconnect broooo....");

                        // Logs out of server
                        ftpClient.logout();
                        ftpClient.disconnect();
                    }
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    };
    ScheduledExecutorService executor = Executors.newScheduledThreadPool(1);
    executor.scheduleAtFixedRate(helloRunnable, 0, 3, TimeUnit.SECONDS);
  } catch (NullPointerException e){
    System.out.println("No data populated.");
  }
}
// Show the Server Reply on Client Request
private static void showServerReply(FTPClient ftpClient) {
  String[] replies = ftpClient.getReplyStrings();
  if (replies != null && replies.length > 0) {
    for (String aReply : replies) {
      System.out.println("SERVER: " + aReply);
    }
  }
}
```

12.1.4.5 Remaining code from Java

The reader may open the .ino, .java, and .cs scripts within the archive or simply open up the whole packages within the designated platforms to see the complete versions of the programs. As previously described, the complete system cannot run without the Arduino and sensors attached. However the visualisation may simply function with only the JSON files as primarily intended.

12.1.4.5.1 Analyser.java

```
package com.analysers;
import beads.*;
import processing.core.PApplet;
import org.json.*;
import java.util.*;

public class Analyser {
   PApplet parent;

   // how many peaks to track and resynth
   int numPeaks = 32;

   ArrayList<String> flcHumanInput;
   ArrayList<ArrayList<String>> acquiredData;

   Random randomGenerator;
   AudioContext ac;
   Gain masterGain;
   PowerSpectrum ps;
   SpectralPeaks sp;
   Gain[] g;
   Glide[] gainGlide;
   Glide[] frequencyGlide;
   WavePlayer[] wp;
   SamplePlayer player;

   // Min and Max ranges for HR and GSR
   int heartRateMinimum;
   int heartRateMaximum;
   int gsrMinimum;
   int gsrMaximum;

   // Randomly generated HR and GR input
   int heartRateInput;
   int gsrInput;

   int time;
   int tickCounter = 1;
   float counter = 3000;
   float meanFrequency = 400.0f;
   int fore;
   int back;

   Analyser(PApplet p) {
      parent = p;

      // Set the background and foreground colours
```

```
        fore = parent.color(255, 255, 255);
        back = parent.color(0, 0, 0);

        // We need to store the multiple samples of index, frequency and
        // amplitude for sound.
        acquiredData = new ArrayList<ArrayList<String>>();

        // We need to store the multiple samples of index, GSR and HR for
        // human-response.
        flcHumanInput = new ArrayList<String>();

        // Random Generator for the HR and GSR dummy data
        randomGenerator = new Random();

        // set up the parent AudioContext object
        ac = new AudioContext();

        // set up a master gain object
        masterGain = new Gain(ac, 2, 0.5f);
        ac.out.addInput(masterGain);

        try {
            // load up a new SamplePlayer using an included audio file
            player = new SamplePlayer(ac, new Sample(parent.sketchPath("") +
"src/com/analysers/CombinedSong_v2.mp3")); //limborockgamedevv2.mp3"));//drumroll.mp3"));
            // connect the SamplePlayer to the master Gain
            masterGain.addInput(player);
        } catch (Exception e) {
            // if there is an error, print the steps that got us to
            // that error
            e.printStackTrace();
        }

        frequencyGlide = new Glide[numPeaks];
        wp = new WavePlayer[numPeaks];
        g = new Gain[numPeaks];
        gainGlide = new Glide[numPeaks];

        for (int i = 0; i < numPeaks; i++) {
            // set up the WavePlayer and the Glides that will control
            // its frequency and gain
            frequencyGlide[i] = new Glide(ac, 440, 1);
            wp[i] = new WavePlayer(ac, frequencyGlide[i], Buffer.SINE);
            gainGlide[i] = new Glide(ac, 0.0f, 1);
            g[i] = new Gain(ac, 1, gainGlide[i]);

            // connect the WavePlayer to the master gain
            g[i].addInput(wp[i]);
            masterGain.addInput(g[i]);
        }

        // in this block of code, we build an analysis chain
        // the ShortFrameSegmenter breaks the audio into short,
        // discrete chunks
        ShortFrameSegmenter sfs = new ShortFrameSegmenter(ac);

        // connect the microphone input to the ShortFrameSegmenter
        // sfs.addInput(microphoneIn);
```

```
        sfs.addInput(player);

        // the FFT transforms that into frequency domain data
        FFT fft = new FFT();

        // connect the ShortFramSegmenter object to the FFT
        sfs.addListener(fft);

        // the PowerSpectrum turns the raw FFT output into proper
        // audio data
        ps = new PowerSpectrum();

        // connect the FFT to the PowerSpectrum
        fft.addListener(ps);

        // the SpectralPeaks object stores the N highest Peaks
        sp = new SpectralPeaks(ac, numPeaks);

        // connect the PowerSpectrum to the Frequency object
        ps.addListener(sp);

        // list the frame segmenter as a dependent, so that the
        // AudioContext knows when to update it
        ac.out.addDependent(sfs);

        // start processing audio
        ac.start();

        // Start timer
        time = parent.millis();
    }

    /**
     * Timer reference:
     * http://stackoverflow.com/questions/12417937/create-a-simple-countdown-in-
     * processing
     */
    public void startAnalysis(double heartRateInputSample, double gsrInputSample) {
        ArrayList<String> flcSoundInput = new ArrayList<String>();

        parent.stroke(fore);

        if ((parent.millis() - time) >= counter) {
            float[][] features = sp.getFeatures();

            for (int i = 0; i < numPeaks; i++) {
                // Store Frequency, Amplitude, HR and GSR input
                double amplitude = log10(features[i][1]);

                flcSoundInput.add(tickCounter + "," + i + "," + features[i][0] + "," + Math.abs(amplitude)
+ "," + (double)heartRateInputSample + "," + (double)gsrInputSample);
            }

            CsvFileWriter.appendCsvFile("hrGSRsamples.txt",
"hr,"+heartRateInputSample+",gsr,"+gsrInputSample);
            acquiredData.add(flcSoundInput);

            // Increase the batch number and reset.
```

```
            ++tickCounter;
            time = parent.millis();
        }
    }

    // calculate base-10 logarithm of a number
    double log10(float x) {
      return (Math.log(x) / Math.log(10));
    }

    // Get the set of data
    public ArrayList<ArrayList<String>> getAcquiredData() {
        return acquiredData;
    }

    // Check if song is playing or complete
    public boolean isSongDone() {
        Sample playerSample = player.getSample();

        if (playerSample.getLength() - player.getPosition() <= 0.0) {
            return true;
        }
        return false;
    }
}
```

12.1.4.5.2 CsvFileWriter.java

```
package com.analysers;

import java.io.*;
import java.util.ArrayList;

public class CsvFileWriter {
    //Separator used in CSV file
    private static final String NEW_LINE_SEPARATOR = "\n";

    public static void appendCsvFile(String fileName, String data) {
        FileWriter fileWriter = null;

        try {
            fileWriter = new FileWriter(fileName, true);

            // Save data to file
            fileWriter.append(data);

            //Add a new line separator after the header
            fileWriter.append(NEW_LINE_SEPARATOR);

        } catch (Exception e) {
            System.out.println("Error when appending CsvFileWriter !!!");
            e.printStackTrace();
        } finally {
            try {
                fileWriter.flush();
                fileWriter.close();
```

```
            } catch (IOException e) {
                System.out.println("Error while flushing/closing fileWriter !!!");
                e.printStackTrace();
            }
        }
    }

}
```

12.1.4.5.3 DataContainer.java

```
package com.analysers;

import java.util.ArrayList;

import org.json.JSONObject;

public class DataContainer {
//  ArrayList<String> extractedData;
//  ArrayList<String> clonedData;
    ArrayList<ArrayList<String>> extractedData;
    ArrayList<ArrayList<String>> clonedData;
    ArrayList<String> hrList;
    ArrayList<String> gsrList;

    JSONObject clonedJSONObject;
    private boolean isSongPlaying;
    int chartCounter;

    public DataContainer() {
        isSongPlaying = false;
        chartCounter = 0;
//      hrList = new ArrayList<String>();
//      gsrList = new ArrayList<String>();
    }

    public int getChartCounter() {
        return chartCounter;
    }

    public void setChartCounter(int newValue) {
        this.chartCounter = newValue;
    }
    public ArrayList<String> getHrList() {
        return hrList;
    }

    public void setHrList(ArrayList<String> hrList) {
        this.hrList = hrList;
    }

    public ArrayList<String> getGsrList() {
        return gsrList;
    }

    public void setGsrList(ArrayList<String> gsrList) {
```

```
        this.gsrList = gsrList;
    }

    public boolean getSongPlaying() {
        return isSongPlaying;
    }

    public void setSongPlaying(boolean isSongPlaying) {
        this.isSongPlaying = isSongPlaying;
    }

    public DataContainer(ArrayList<ArrayList<String>> extractedData) {
        this.extractedData = extractedData;
    }

    public DataContainer(ArrayList<ArrayList<String>> extractedData,
ArrayList<ArrayList<String>> clonedData) {
        this.extractedData = extractedData;
        this.clonedData = clonedData;
    }

    public ArrayList<ArrayList<String>> getExtractedData() {
        return extractedData;
    }

    public void setExtractedData(ArrayList<ArrayList<String>> extractedData) {
        this.extractedData = extractedData;
    }

    public ArrayList<ArrayList<String>> getClonedData() {
        return clonedData;
    }

    public void setClonedData(ArrayList<ArrayList<String>> clonedData) {
        this.clonedData = clonedData;
    }

    public JSONObject getClonedJSONObject() {
        return clonedJSONObject;
    }

    public void setClonedJSONObject(JSONObject clonedJSONObject) {
        this.clonedJSONObject = clonedJSONObject;
    }

    @SuppressWarnings("unchecked")
    public void cloneExtractedData() {
        this.clonedData = this.extractedData;
    }
}
```

12.1.4.5.4 ControllerInputExtractor.java – remaining code that was not listed previously

```
   // Fuzzy Logic Controller - Given the elicited data, Fuzzify it.
   public static JSONObject runFuzzyLogic(int pivot, ArrayList<String> dataList,
ArrayList<ArrayList<String>> setOfData) {
      JSONObject dataObject = new JSONObject();
      ArrayList<String> variables = new ArrayList<String>();
      // Error while loading?
      if (fis == null) {
         System.err.println("Cannot load file: '" + fileName + "'");
         return dataObject;
      }

      // Fuzzify data and store them in new array list for further wrapping
      for (int k = 0 + pivot; k < dataList.size(); k++) {
         String[] line = dataList.get(k).split(",");

         // Set inputs
         fis.setVariable("frequency",Double.parseDouble(line[2]));
         fis.setVariable("amplitude", Double.parseDouble(line[3]));
         fis.setVariable("heartrate", Double.parseDouble(line[4]));
         fis.setVariable("galvanic", Double.parseDouble(line[5]));

         // Show output variable's chart
         Variable colour = fis.getVariable("colour");
         Variable size = fis.getVariable("size");
         Variable pulse = fis.getVariable("pulse");
         Variable speed = fis.getVariable("speed");

         // Fuzzify the elicited data
         fis.evaluate();

         // Get the Fuzzified Values
         double colourDouble = fis.getVariable("colour").getValue();
         double sizeDouble = fis.getVariable("size").getValue();
         double pulseDouble = fis.getVariable("pulse").getValue();
         double speedDouble = fis.getVariable("speed").getValue();

         // Add them to new list to store in JSONObject
         variables.add(line[0] + ", " + line[1] + ", " + colourDouble + ", " + sizeDouble + ", " +
pulseDouble + ", " + speedDouble);
      }

      // Wrap the fuzzified data into JSONObjects and add to list
      for (int k = 0 + pivot; k < variables.size(); k++) {
         JSONObject dataObjectValue = new JSONObject();
         String[] lines = variables.get(k).split(",");
         if (k <= dataList.size()-1) {
            pivot = Integer.parseInt(lines[0]);
         }

         dataObjectValue.put("colour", lines[2]);
         dataObjectValue.put("size", lines[3]);
         dataObjectValue.put("pulse", lines[4]);
         dataObjectValue.put("speed", lines[5]);
         dataObject.put(lines[1], dataObjectValue);
      }
```

```
    return dataObject;
  }

  public void settings() {
      size(760, 480);
  }

  public void setup() {
    myPort = new SerialPort("COM4");
    // used for the "calibrating" text displayed at the beginning
    f = createFont("Arial",24);
    // used for the avgBPM display on screen
    f2 = createFont("Arial",60);

    // Initialise the BPM array
    for(int i = 0; i < totalBeats; i++) {
      BPM[i] = 0;
    }

    // Initialise the GSR array
    for(int i = 0; i < totalGSRSamples; i++) {
      GSR[i] = 0;
    }

    // Start Serial communication with the Arduino using a baud rate of 9600
    try {
        myPort.openPort();
        myPort.setParams(myPort.BAUDRATE_9600,                    myPort.DATABITS_8,
myPort.STOPBITS_1, myPort.PARITY_NONE);
        myPort.setFlowControlMode(myPort.FLOWCONTROL_RTSCTS_IN                    |
myPort.FLOWCONTROL_RTSCTS_OUT);
        myPort.addEventListener(new PortReader(), myPort.MASK_RXCHAR);
    } catch (SerialPortException e) {
        e.printStackTrace();
    }

    analyser = new Analyser(this);

    // Setup the List of elicited data lists
    data = new ArrayList<ArrayList<String>>();

    // Instantiate the container class passing it the set to be stored
    dataContainerClass = new DataContainer(data);

    // Song is playing by default
    dataContainerClass.setSongPlaying(false);
  }

  int i = 0;
  double inputHRsample, inputGSRsample;
  public void draw() {
     background(0);
     drawBPM();
     drawGSR();
     analyser.startAnalysis(avgBPM, avgGSRValue);

     // Get the song state (playing or stopped)
```

```
        dataContainerClass.setSongPlaying(analyser.isSongDone());

        // Add the ArrayList of 32 pieces of data to the list
        setData(analyser.getAcquiredData());

        // Store a copy of the extracted elicited Data to be processed
        dataContainerClass.setExtractedData(data);

        // Quit the Application if song ends.
        if (dataContainerClass.getSongPlaying()) {
            println("PROCESSING CHECK - Is song done: " + dataContainerClass.getSongPlaying());
            println("Final Data Copy Size: " + data.size());

            // Sleep for 6 seconds to allow all data to be sent to server
            try { Thread.sleep(6000); }
            catch(InterruptedException e) { };

            // Exit the program if song and 6 seconds have ended
            exit();
        }
    }

    ArrayList<Double> gsrSamples = new ArrayList<Double>();
    void drawGSR() {
      // Reset the sumGSR variable
      sumGSR = 0;
      // Reset the avgGSR variable
      avgGSRValue = 0;

      boolean userAbsent = false;

      for(int i = 1; i<totalGSRSamples; i++){
        // Sum all of the BPM values in the BPM array.
        sumGSR = sumGSR + GSR[i-1];
        // If any GSR values are equal to 390 or 391, then set the calibrating variable to true.
        if(GSR[i-1] >= 379) {//GSR[i-1] == 389 || GSR[i-1] == 390 || GSR[i-1] == 391) {
          // This will be used later to display "calibrating" on the screen.
          userAbsent = true;
        }
      }
      // Calculate the average BPM from all BPM values
      avgGSRValue = sumGSR/(totalGSRSamples-1);

      fill(255);
      if(userAbsent){
        textFont(f);
        // If the calibrating variable is TRUE, then display the word "Calibrating" on screen
        text("User Absent", (2*width)/5, (height/5));
        // Change the fill and stroke to black (0) so that other text is "hidden" while calibrating variable
is TRUE
        fill(0);
        stroke(0);
      } else {
        textFont(f2);
        // If the calibrating variable is FALSE, then display the avgBPM variable on screen
        text(avgGSRValue, (2*width)/5, (height/5));
        // Change the stroke to white (255) to show the
        // white line underlying the word BPM.
```

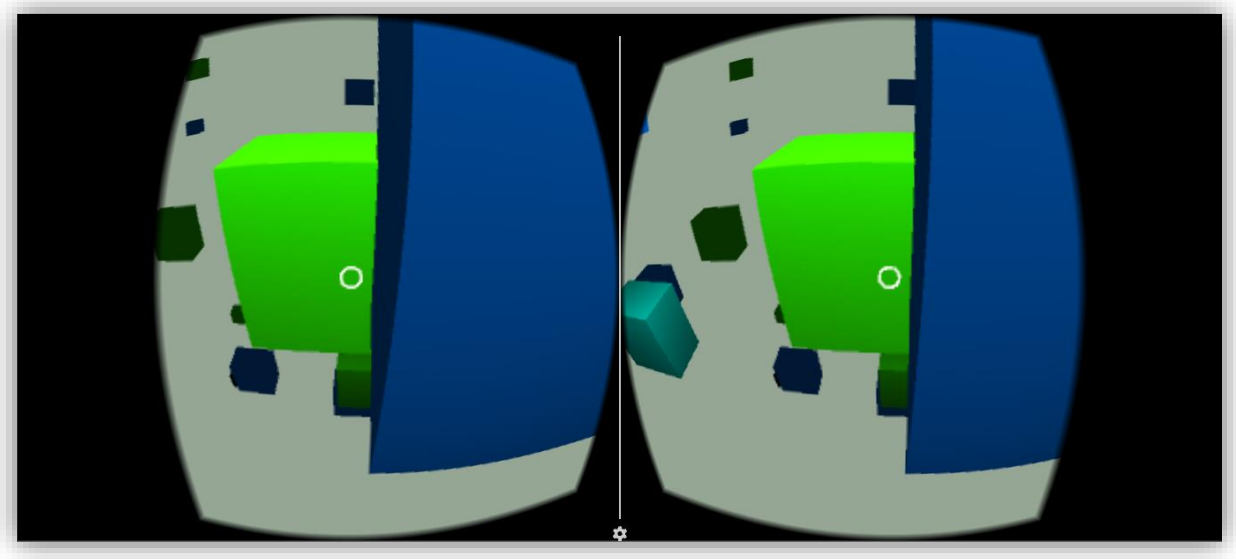
```
      stroke(255);
    }

    textFont(f);
    // This will display the underlined word "BPM" when calibrating variable is FALSE.
    text("GSR", (41*width)/100, (height/11));
    line((40*width)/100, (height/10),(44*width)/100, (height/10));
    stroke(0);
  }

  ArrayList<Double> hrSamples = new ArrayList<Double>();
  void drawBPM() {
   // Reset the sumBPM variable
   sumBPM = 0;
   // Reset the avgBPM variable
   avgBPM = 0;
   // calibrating: this boolean variable is used
   //            to control when the avgBPM is displayed to screen
   boolean calibrating = false;

   for(int i=1; i<totalBeats; i++){
    // Sum all of the BPM values in the BPM array.
    sumBPM = sumBPM + BPM[i-1];
    // If any BPM values are equal to 0, then set the calibrating variable to true.
    if(BPM[i-1]<1) {
     // This will be used later to display "calibrating" on the screen.
     calibrating = true;
    }
   }
   // Calculate the average BPM from all BPM values
   avgBPM = (sumBPM/(totalBeats-1))*2;

   fill(255);
   if(calibrating){
    textFont(f);
    // If the calibrating variable is TRUE, then display the word "Calibrating" on screen
    text("Calibrating", (4*width)/5, (height/5));
    // Change the fill and stroke to black (0) so that other text is "hidden" while calibrating variable
is TRUE
    fill(0);
    stroke(0);
   } else {
    textFont(f2);
    // If the calibrating variable is FALSE, then display the avgBPM variable on screen
    text(avgBPM, (4*width)/5, (height/5));
    // Change the stroke to white (255) to show the
    // white line underlying the word BPM.
    stroke(255);
   }

    textFont(f);
    // This will display the underlined word "BPM" when calibrating variable is FALSE.
    text("BPM", (82*width)/100, (height/11));
    line((80*width)/100, (height/10),(88*width)/100, (height/10));
    stroke(0);
  }
```

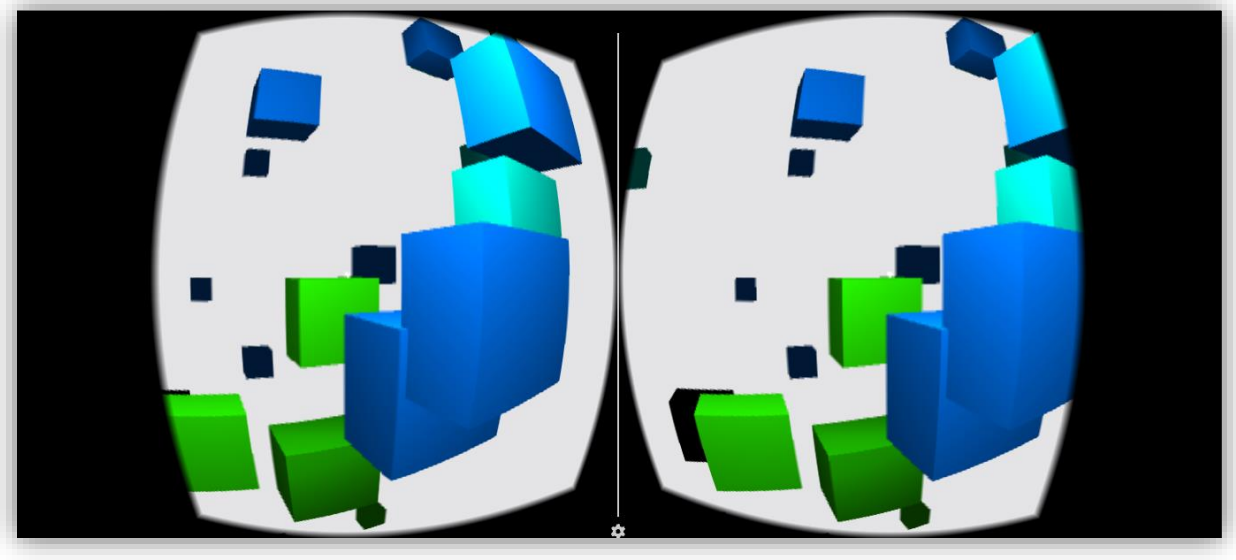
```
  // This method is used to calculate the Beats per Minute (BPM) and to store the last 10 BPMs into the BPM[] array.
  void triggerGSR(int time, int actualValue) {
    // totalTime = the current beat time minus the last time there was a beat.
    gsrTotalTime = time - gsrLastTime;
    // Set the lastTime variable to the current "time" for the next round of calculations.
    gsrLastTime = time;
    // Calculate BPM from the totalTime. 60000 = 1 minute.
    GSR[gsrCounter] = actualValue;
    // Increment the beatCounter
    gsrCounter++;

    // Reset the beatCounter when the total number of BPMs have been stored into the BPM[] array.
    if (gsrCounter>totalGSRSamples-1) {
      // This allows us to keep the last 10 BPM calculations at all times.
      gsrCounter = 0;
    }
  }

  // This method is used to calculate the Beats per Minute (BPM) and to store the last 10 BPMs into the BPM[] array.
  void triggerHR(int time) {
    // totalTime = the current beat time minus the last time there was a beat.
    totalTime = time - lastTime;
    // Set the lastTime variable to the current "time" for the next round of calculations.
    lastTime = time;
    // Calculate BPM from the totalTime. 60000 = 1 minute.
    BPM[beatCounter] = 60000/totalTime;
    // Increment the beatCounter
    beatCounter++;

    // Reset the beatCounter when the total number of BPMs have been stored into the BPM[] array.
    if (beatCounter>totalBeats-1) {
      // This allows us to keep the last 10 BPM calculations at all times.
      beatCounter=0;
    }
  }

  public Analyser getAnalyser() {
    return analyser;
  }

  public void setAnalyser(Analyser analyser) {
    this.analyser = analyser;
  }

  public ArrayList<ArrayList<String>> getDataList() {
    return data;
  }

  public void setData (ArrayList<ArrayList<String>> data) {
    this.data = data;
  }
```

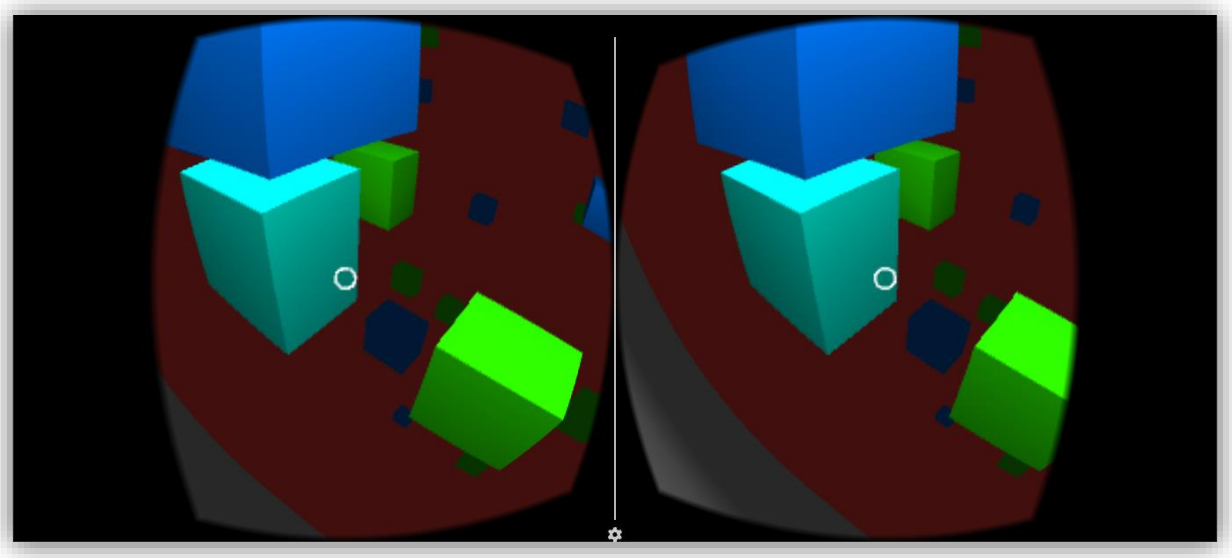
Computer and Devices used for Output Module

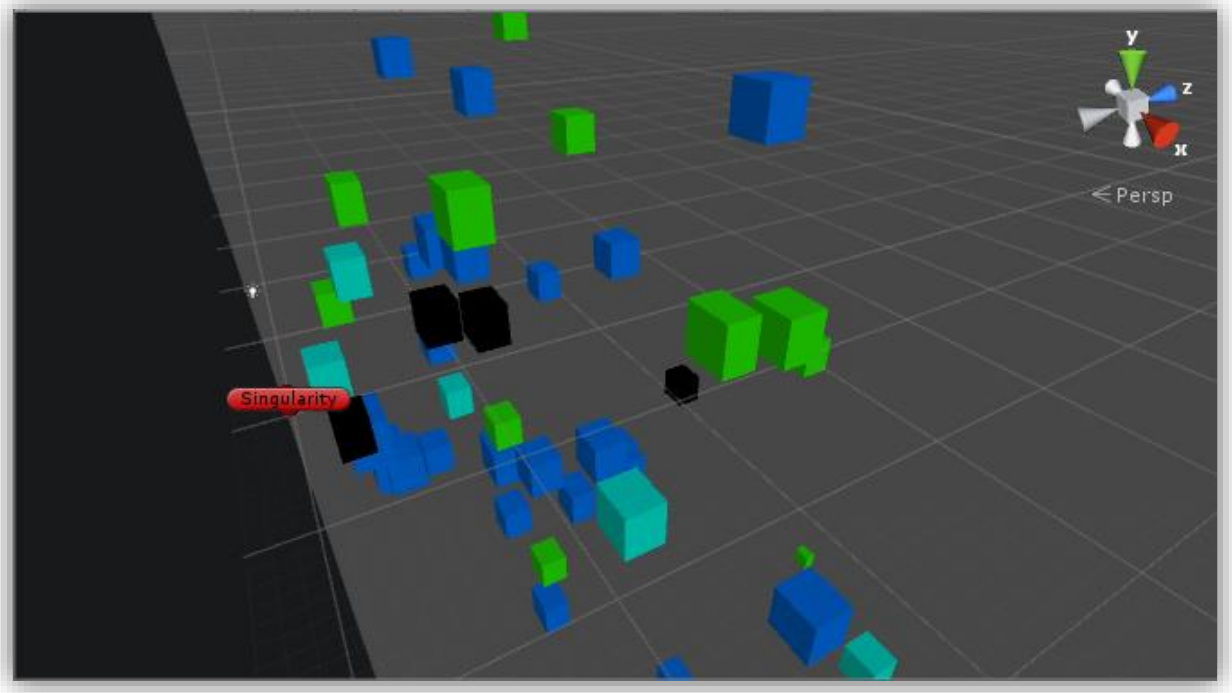
- Laptop – Fujitsu Lifebook A531
- Mobile Device – OnePlus One 64 GB Sandstone Edition

12.1.5 Unity3D snapshots








y
z
x
Persp
Singularity

12.1.6 **Visualisation Components**

12.1.6.1 Unity3D Program Scripts

GetURL.cs

```
using UnityEngine;
using System.Collections;
using System.Collections.Generic;
using System.Net;
using System.ComponentModel;
using System.IO;
using LitJson;

public class GetURL : MonoBehaviour {
    private int pivot = 0;
    private bool isRunning, stopSearching, foundFirstURL, foundRequestedURL;
    private string firstUrl = "http://websitename.domainname/data/1_fuzzySet.json", url,
previousUrl, newUrl;

    private float timeLeft = 10.0f;
    public DataClassContainer dataClassContainer;

    // Create an instance of DataClassContainer should it not exist
    void Awake() {
        if (DataClassContainer.instance == null)
            Instantiate(dataClassContainer);
    }

    void Start () {
        // Establish connection with FTP Server and Login using credentials
        FtpWebRequest request =
(FtpWebRequest)WebRequest.Create("ftp://websitename.domainname/data/");
        request.Credentials = new NetworkCredential("<username>", "<password>");
        request.Method = "NLST";

        // Set the states for iterative search
        isRunning = false;
        stopSearching = false;
        foundFirstURL = false;
        foundRequestedURL = false;
        pivot = 1;

    }

    void Update() {
        timeLeft -= Time.deltaTime;

        if (timeLeft <= 0 && !stopSearching) {
            timeLeft = 10.0f;

            // If we are still on the first URL (i.e. first url not found)
            if (!foundFirstURL) {
                // Try check if the URL exists
                foundFirstURL = URLExistsNoRedirect (firstUrl);

                // If it exists - Process request
                if (foundFirstURL) {
```

```
                // Keep reference to previous URL
                previousUrl = firstUrl;

                // Request connection to URL
                StartCoroutine (WaitForRequest (firstUrl));

                // Allow for more files to be retrieved
                pivot = pivot + 1;

                // We have now found the first URL
                foundFirstURL = true;
            } else {
                pivot = pivot + 1;
                previousUrl = firstUrl;
                foundFirstURL = true;
            }

          // If the first search is complete
        } else if (!stopSearching && foundFirstURL && !foundRequestedURL) {
            // Try next URL
            string tryNewUrl = "http://websitename.domainname/data/" + pivot + "_fuzzySet.json";
            foundRequestedURL = URLExistsNoRedirect (tryNewUrl);

            // If the new URL is not like the previous one and it exists, download the file.
            if (!(previousUrl.Equals (tryNewUrl)) && foundRequestedURL) {
                newUrl = tryNewUrl;
                previousUrl = newUrl;

                StartCoroutine (WaitForRequest (newUrl));

                // Go to next file
                pivot = pivot + 1;
                foundRequestedURL = false;
                stopSearching = false;

            // URL not found, go to next URL
            } else if (!foundRequestedURL) {
                pivot = pivot + 1;
                previousUrl = tryNewUrl;
                foundRequestedURL = false;

            // No more files exist on server
            } else {
                foundRequestedURL = true;
                stopSearching = true;
            }
        }

        // Indicate that we have stopped processing data from server
        if (stopSearching && foundFirstURL && foundRequestedURL) {
            Debug.Log ("We're updating but not the WebRequest...!");
        }
    }
}

// Coroutine for downloading files from server
IEnumerator WaitForRequest (string url) {
    WWW www = new WWW (url);
```

```
        yield return www;

        // check for errors
        if (www.error == null) {

            string jsonString = www.text;

            // Decode and store the content of the file in the DataClassContainer
            ReadJSON(jsonString);
        } else {
            Debug.Log("WWW Error: " + www.error);
        }
    }

    // Store the contents of each file in an arraylist
    static void ReadJSON(string jsStr) {
        // Get the JSONReader Parser
        JsonReader reader = new JsonReader(jsStr);

        // States to parse packets from files
        bool restart = false, startSavingPropertyName = false, startSavingPacket = false,
firstRowOfPacket = false, firstObjectEnd = false, lastObjectEnd = false, saveClr = false, saveSz
= false, savePls = false, saveSpd = false;
        int propertyCounter = 0, previousCount = 0;

        // Start parsing JSON Content - for every existing packet
        while(reader.Read()) {

            // If we have parsed the whole File.
            if (restart) {
                previousCount = 0;
                propertyCounter = 0;
                firstRowOfPacket = false;
                startSavingPacket = false;
                startSavingPropertyName = false;
                saveClr = false;
                saveSz = false;
                savePls = false;
                saveSpd = false;
                DataClassContainer.instance.resetProperties();
                DataClassContainer.instance.resetPacket();
                firstObjectEnd = false;
                lastObjectEnd = false;
                restart = false;
            }

            // Reader starts from ObjectStart indicated by '{'
            string type = reader.Value != null ? reader.Value.GetType().ToString() : "";

            if (!restart) {
                // If this is the first line of the JSON File
                if (reader.Token.ToString().Equals("ObjectStart") && !firstRowOfPacket &&
!startSavingPropertyName && !startSavingPacket) {
                    firstRowOfPacket = true;

                // Get the row number and store it
                } else if (reader.Token.ToString().Equals("PropertyName") && firstRowOfPacket &&
!startSavingPropertyName && !startSavingPacket) {
```

```
                startSavingPropertyName = true;
                DataClassContainer.instance.SetPropertyName(reader.Value.ToString());

            // Indicate start of nested object (1 out of 4 main properties)
            } else if (reader.Token.ToString().Equals("ObjectStart") && firstRowOfPacket &&
startSavingPropertyName && !startSavingPacket) {
                startSavingPacket = true;

            // First property - Colour
            }       else        if      (reader.Token.ToString().Equals("PropertyName")       &&
reader.Value.Equals("colour")  &&  startSavingPacket)  {//&&  firstRowOfPacket  &&
!startSavingPropertyName && !startSavingPacket) {
                saveClr = true;

            // Store the colour value
            } else if (saveClr && !saveSz && !savePls && !saveSpd) {
                DataClassContainer.instance.SetColour(reader.Value.ToString());
                saveClr = false;

            // Second property - Size
            }       else        if      (reader.Token.ToString().Equals("PropertyName")       &&
reader.Value.ToString().Equals("size")) {
                saveSz = true;

            // Store the size value
            } else if (!saveClr && saveSz && !savePls && !saveSpd) {
                DataClassContainer.instance.SetSize(reader.Value.ToString());
                saveSz = false;

            // Third property - Pulse
            }       else        if      (reader.Token.ToString().Equals("PropertyName")       &&
reader.Value.ToString().Equals("pulse")) {
                savePls = true;

            // Store the pulse value
            } else if (!saveClr && !saveSz && savePls && !saveSpd) {
                DataClassContainer.instance.SetPulse(reader.Value.ToString());
                savePls = false;

            // Fourth property - Speed
            }       else        if      (reader.Token.ToString().Equals("PropertyName")       &&
reader.Value.ToString().Equals("speed")) {
                saveSpd = true;

            // Store the speed value
            } else if (!saveClr && !saveSz && !savePls && saveSpd) {
                DataClassContainer.instance.SetSpeed(reader.Value.ToString());
                saveSpd = false;
                firstObjectEnd = true;

            // If this is the last row in the packet (end of line of row from all 32 rows)
            } else if (reader.Token.ToString().Equals("ObjectEnd") && firstObjectEnd) {
                propertyCounter += 1;
                string                                  property                                =
DataClassContainer.instance.GetCombineProperties(propertyCounter);
                DataClassContainer.instance.AddCombinedPropertyToPacket(property);

                startSavingPropertyName = false;
```

```
                startSavingPacket = false;
                firstObjectEnd = false;
                lastObjectEnd = true;

                // If this is true, the JSON object is closing.  Store the whole packet in a list.
            } else if (reader.Token.ToString().Equals("ObjectEnd") && !firstObjectEnd && lastObjectEnd) {
                DataClassContainer.instance.AddPacketToTotalPacketsReceived();
                restart = true;
            }
        }

        // Reader ends with ObjectEnd indicated by '}'
    }

}

/// <summary>
/// URLs the exists.
/// Source: http://stackoverflow.com/questions/1979915/can-i-check-if-a-file-exists-at-a-url
/// </summary>
/// <returns> true, if URL exists, false otherwise.</returns>
/// <param name="url">URL.</param>
static public bool URLExists (string url) {
    bool result = false;

    WebRequest webRequest = WebRequest.Create(url);
    webRequest.Timeout = 60000; // milliseconds
    webRequest.Method = "HEAD";
    HttpWebResponse response = null;

    try {
        response = (HttpWebResponse) webRequest.GetResponse();
        result = true;
    } catch(WebException webException) {
        Debug.Log (url + " doesn't exist: " + webException.Message);
    } finally {
        if (response != null) {
            response.Close();
        }
    }
    return result;
}

/// <summary>
/// Check if URL exists without automatic redirect
/// Source: http://stackoverflow.com/questions/1979915/can-i-check-if-a-file-exists-at-a-url
/// Author: LastEnd - 1st August, 2014 at 21:33
/// </summary>
/// <returns> true, if URL exists, false otherwise.</returns>
/// <param name="url">URL.</param>
static public bool URLExistsNoRedirect (string url) {
    bool result = false;

    HttpWebResponse responseR = null;
    var webRequest = (HttpWebRequest)WebRequest.Create(url);
    webRequest.Method = "HEAD";
    webRequest.Timeout = 60000; // milliseconds
```

```
        webRequest.AllowAutoRedirect = false;


        try {
            responseR = (HttpWebResponse) webRequest.GetResponse();
            result = responseR.StatusCode == HttpStatusCode.OK;
        } catch(WebException webException) {
            Debug.Log (url + " doesn't exist: " + webException.Message);
        } finally {
            if (responseR != null) {
                responseR.Close();
            }
        }
        return result;
    }
}
```

JSON Parser and Storage of data in Singleton Class

List the code and logic behind the use of the Singleton class.

```
using UnityEngine;
using System.Collections;
using System.Collections.Generic;
/// <summary>
/// Data class container singleton for transferring of JSON data
///     Source:     https://unity3d.com/learn/tutorials/projects/2d-roguelike-tutorial/writing-game-manager
/// </summary>
public class DataClassContainer : MonoBehaviour {
    public static DataClassContainer instance = null;
    private List<List<string>> totalPackets;
    private List<string> packet;
    private string propertyName, colour, size, pulse, speed;
    private string combinedProperty;

    void Awake() {
        if (instance == null)
            instance = this;
        else if (instance != this)
            Destroy (gameObject);

        DontDestroyOnLoad(gameObject);
    }

    void Start() {
        packet = new List<string>();
        totalPackets = new List<List<string>> ();
        propertyName = "";
        colour = "";
        size = "";
        pulse = "";
        speed = "";
        combinedProperty = "";
    }

    public List<List<string>> GetTotalPacketsList() {
        return totalPackets;
    }

    /// <summary>
    /// Resets the packet list.
    /// Use once the user has read one row of the 32 in one file.
    /// </summary>
    public void resetPacket() {
        packet.Clear ();
    }

    /// <summary>
    /// Resets the total packets.
    /// Use only if really neccessary.
    /// </summary>
    public void resetTotalPackets() {
        totalPackets.Clear ();
    }
```

```
/// <summary>
/// Resets all properties.
/// </summary>
public void resetProperties() {
    propertyName = "";
    colour = "";
    size = "";
    pulse = "";
    speed = "";
    combinedProperty = "";
}

public string GetPropertyName() {
    return this.propertyName;
}

public string GetColour() {
    return this.colour;
}

public string GetSize() {
    return this.size;
}

public string GetPulse() {
    return this.pulse;
}

public string GetSpeed() {
    return this.speed;
}
/// <summary>
/// Adds the combined property to packet.
/// Use when a row from all 32 rows in one file is combined.
/// </summary>
/// <param name="combinedProperty">Combined property.</param>
public void AddCombinedPropertyToPacket(string combinedProperty) {
    this.combinedProperty = combinedProperty;
    packet.Add (this.combinedProperty);
}

/// <summary>
/// Adds the packet to total packets received.
/// Use when we would like to move onto the next row in the file
/// </summary>
public void AddPacketToTotalPacketsReceived() {
    totalPackets.Add (packet);
}

/// <summary>
/// Gets the properties combined.
/// Combine all properties in a row to add to a packet.
/// </summary>
/// <returns>The combine properties.</returns>
/// <param name="propertyName">Property name.</param>
/// <param name="colour">Colour.</param>
/// <param name="size">Size.</param>
```

```
    /// <param name="pulse">Pulse.</param>
    /// <param name="speed">Speed.</param>
    public string GetCombineProperties(string propName, string clr, string sz, string pls, string
spd) {
        string newProperty = "";
        newProperty = propName + ", " + clr + ", " + sz + ", " + pls + ", " + spd;
        return newProperty;
    }

    /// <summary>
    /// Gets the properties combined.
    /// Combine all properties in a row to add to a packet.
    /// </summary>
    /// <returns>The combine properties.</returns>
    public string GetCombineProperties(int propCounter) {
        string newProperty = "";
        newProperty = propCounter + ", " + propertyName + ", " + colour + ", " + size + ", " + pulse + ",
" + speed;
        return newProperty;
    }

    public void SetCombinedProperties(string cmbProp) {
        this.combinedProperty = cmbProp;
    }

    public void SetPropertyName(string val) {
        this.propertyName = val;
    }

    public void SetColour(string clr) {
        this.colour = clr;
    }

    public void SetSize (string sz) {
        this.size = sz;
    }

    public void SetPulse (string pls) {
        this.pulse = pls;
    }

    public void SetSpeed(string spd) {
        this.speed = spd;
    }
}
```

Unity3D GameObjects and other utilities

Explain the use of GameObjects and other utilities provided by Unity for a better understanding for their role in the system.

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;
using System.Collections.Generic;
using System.Linq;

/// <summary>
/// Spawner - The base from which spawnables are created
/// </summary>
public class Spawner : MonoBehaviour {
    public GameObject surface;
    private GameObject spawnArea;
    private float range;
    public float spawnRate;
    public GameObject spawnable;
    public GameObject boundary;
    private BoxCollider boundaryCollider ;

    private Dictionary<int, GameObject> spawnableRegistry;
    private DataClassContainer dataClassContainer;
    private List<List<string>> batchesOfFuzzyData;

    private int currPosInBatchData = 0, currPosInActualData = 0, spawnableCount = 0;

    void Start () {
        spawnableRegistry = new Dictionary<int, GameObject>();
        GameObject dataContainer = GameObject.FindGameObjectWithTag ("Data");
        dataClassContainer = dataContainer.GetComponent<DataClassContainer> ();
        batchesOfFuzzyData = new List<List<string>> ();


        boundary = GameObject.FindGameObjectWithTag("Boundary");
        spawnArea = GameObject.FindGameObjectWithTag("SpawnArea");
        range = spawnArea.GetComponent<SphereCollider>().radius *
spawnArea.transform.localScale.x;
        boundaryCollider = boundary.GetComponent<BoxCollider>();
        InvokeRepeating("Spawn", 0, 3.0f / spawnRate);
    }

    // Update the number of packets retrieved
    void Update() {
        batchesOfFuzzyData = dataClassContainer.GetTotalPacketsList ();
    }

    void Spawn() {
        if (batchesOfFuzzyData.Count > 0) {
            // Increment number of instantiated spawnables
            spawnableCount += 1;

            // Add the new spawnable to the dictionary, along with its ID
            spawnableRegistry.Add (spawnableCount, spawnable);

            // Instantiate spawnable but first apply main properties!
```

```
        (Instantiate (spawnable) as GameObject).GetComponent<Spawnable>
().ApplyAllSpawnableProperties (GetTheSpawnableProperties
());//spawnableRegistry.Values.ElementAtOrDefault(currPosInActualData)));
        }
    }

    // Apply the visual properties to the spawnable game object
    string GetTheSpawnableProperties() {//Object tempSpwn) {
        string[] currentRowData;
        string newColour = "";
        float newShapeSize = 0F, newPulseRate = 0F, newSpeed = 0F;
        Color finalColour;

        if (currPosInActualData > 0) {
            int modulator = Mod (currPosInActualData, 32);

            // If we reached the last row in the packet, increment the file number to decode
            if (modulator == 0 && batchesOfFuzzyData.Count > 0) {
                currPosInBatchData += 1;
                newPulseRate = 0F;
                newSpeed = 0F;
            }
        }

        // If there exists at least 1 file
        if (!(currPosInBatchData >= batchesOfFuzzyData.Count) && batchesOfFuzzyData.Count >=
1) {
            // Get the row data from the decoded JSON string
            currentRowData =
batchesOfFuzzyData[currPosInBatchData][currPosInActualData].ToString().Split(',');
            // Set the spawnable colour
            newColour = colourMapping(float.Parse(currentRowData[2]));

            // Set the spawnable shape size
             newShapeSize = float.Parse(currentRowData[3]);


            // Set the spawnable pulse rate
             newPulseRate = float.Parse(currentRowData[4]);

            // Set the spawnable converging speed
             newSpeed = float.Parse(currentRowData[5]);

            currPosInActualData += 1;
        }
        return newColour+","+newShapeSize+","+newPulseRate+","+newSpeed;
    }

    // Modulator calculation
    int Mod (int a, int b) {
        return (a % b + b) % b;
    }

    // The mapping from Fuzzy Logic Values to colour name for further conversion
    // See Lemon - 5519.5419 Angstroms on http://www.lunarplanner.com/Harmonics/planetary-
harmonics.html
    string colourMapping(float colourValue) {
        string mappedColour = "";
```

```
        if (colourValue >= 3683.8492 && colourValue <= 3902.8999) {
            mappedColour = "nearuv";
        } else if (colourValue >= 3902.9001 && colourValue <= 4134.9824) {
            mappedColour = "deepviolet";
        } else if (colourValue >= 4134.9825 && colourValue <= 4380.8603) {
            mappedColour = "violet";
        } else if (colourValue >= 4380.8603 && colourValue <= 4641.3564) {
            mappedColour = "blueind";
        } else if (colourValue >= 4641.3565 && colourValue <= 4917.3520) {
            mappedColour = "blue";
        } else if (colourValue >= 4917.3521 && colourValue <= 5212.2521) {
            mappedColour = "cyan";
        } else if (colourValue >= 5212.2522 && colourValue <= 5519.5418) {
            mappedColour = "green";
        } else if (colourValue >= 5519.5419 && colourValue <= 5847.7442) {
            mappedColour = "lemon";
        } else if (colourValue >= 5847.7443 && colourValue <= 6195.4724) {
            mappedColour = "yellow";
        } else if (colourValue >= 6195.4725 && colourValue <= 6583.8696) {
            mappedColour = "orange";
        } else if (colourValue >= 6583.8697 && colourValue <= 6954.1726) {
            mappedColour = "redorange";
        } else if (colourValue >= 6954.1727 && colourValue <= 7367.6983) {
            mappedColour = "red";
        } else if (colourValue >= 7367.6984 && colourValue <= 7805.8002) {
            mappedColour = "deepred";
        } else if (colourValue >= 7805.8003) {
            mappedColour = "ifr";
        }
        return mappedColour;
    }

    // Starting position for the spawnables, make sure it is within the dome
    public Vector3 startPosition() {
        Vector3 position = (Random.insideUnitSphere * range) + spawnArea.transform.position;
        while (IsWithinBoundaries(position, boundaryCollider))
        {
            position = (Random.insideUnitSphere * range) + spawnArea.transform.position;
        }

        return position;
    }
        //Is point within circle outside of the "forbidden area" defined by collider's area
    bool IsWithinBoundaries(Vector3 point, BoxCollider box)
    {
        point = box.transform.InverseTransformPoint(point) - box.center;

        float halfX = (box.size.x * 0.5f);
        float halfY = (box.size.y * 0.5f);
        float halfZ = (box.size.z * 0.5f);
        if (point.x < halfX && point.x > -halfX &&
            point.y < halfY && point.y > -halfY &&
            point.z < halfZ && point.z > -halfZ)
            return true;
        else
            return false;
    }
}
```

Mapping data for the visualisation

The code that handles the mapping of properties onto the spawnable instance.

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;
using System.Collections.Generic;
using Assets;
using DG.Tweening;


//SPAWNABLE PULSE RATE: HEART RATE
//SPAWNABLE COLOR: FREQUENCY
//SPAWNABLE SPEED: GSR
//SPAWNABLE SIZE: AMPLITUDE

public class Spawnable : MonoBehaviour
{

    public Color color; //{ get; set; }
    public int maximumCollisions = 10;
    private Transform endTransform;

    private Vector3 startMarker;
    private Renderer renderer;
    private Vector3 endMarker;

    private Tween pulse;
    private float fauxPulseRateTemp;
    private int totalCollisions = 0;

    private bool runSingularity = true;

    //maximum and minimum pulse size (GSR)
    public Vector3 minScale = new Vector3(0.5f, 0.5f, 0.5f);
    public Vector3 maxScale = new Vector3(1.5f, 1.5f, 0.5f);

    //Spawnable movement speed (AMPLITUDE)
    public float speed = 1.0F;

    private ThesisCamera cameraScript;
    private float timeLeftToCheckPulseUpdate;// = 1.0F;

    void Start() {
        // Initialise DOTween
        DOTween.Init();

        //Find and instantiate variables
        //When performance is spiky, set these as public variables and and assign them to the prefab
manually
        GameObject spawnerObject = GameObject.FindGameObjectWithTag("Spawner");
        endTransform = GameObject.FindGameObjectWithTag("EndTransform").transform;

        // Get the spawner and main camera script
        Spawner spawner = spawnerObject.GetComponent<Spawner>();
        GameObject mainCamera = GameObject.FindGameObjectWithTag("MainCamera");
        cameraScript = mainCamera.GetComponent<ThesisCamera>();
```

```
        renderer = gameObject.GetComponentInChildren<Renderer>();

        //Set start and end position, distance, and start time
        startMarker = spawner.startPosition();
        transform.position = startMarker;
        endMarker = endTransform.position;
    }

    //Sequence of tweens powered by DOTween
    void ScaleTween() {
        Sequence sequence = DOTween.Sequence ();
        sequence.Append (transform.DOScale (maxScale, 0.75f * fauxPulseRateTemp));
        sequence.Append (transform.DOScale (minScale, 0.25f * fauxPulseRateTemp));
    }

    void Update() {
        // Set material color
        renderer.material.SetColor("_Color", color);

        // Ensure all spawnables move towards user
        if (runSingularity) {
            if (totalCollisions <= maximumCollisions)
                Singularity();
            else
            {
                Destroy(gameObject);
            }
        }
    }

    //Destroy if objects bump in to one another excessively or if reach end transform
    void OnCollisionEnter(Collision collision) {
        if (collision.gameObject.tag == "EndTransform")
        {
            if (cameraScript != null) {
                cameraScript.Flash();
            }
            Destroy(gameObject);
        }
        totalCollisions++;
        if (runSingularity)
        {
            runSingularity = false;
            Invoke("ToggleSingularity", 3);
        }

    }

    //Movement towards singularity gameobject
    void Singularity() {
        if (transform.position != endMarker)
        {
            float step = speed * Time.deltaTime;
            transform.position = Vector3.MoveTowards(transform.position, endMarker, step);
        }
    }
```

```
public Color GetColour() {
    return this.color;
}

// Pass the Frequency data from Server
public void SetColour(Color tempClr) {
    this.color = tempClr;
}

// Pass the Amplitude data from server
public void SetShapeSize(float incSize) {
    minScale.x = incSize;
    minScale.y = incSize;
    minScale.z = incSize;
    maxScale.x = incSize + 1;
    maxScale.y = incSize + 1;
    maxScale.z = incSize;
}

// Pass the HR ratedata from Server
public float GetFauxPulseRate() {
    return fauxPulseRateTemp;
}

// Pass the HR ratedata from Server
public void SetFauxPulseRate(float fauxPulseRate) {
    this.fauxPulseRateTemp = fauxPulseRate;
}

// Save the GSR data from Server
public void SetSpeed(float newSpeed) {
    speed = 1.0F * newSpeed;
}

public float GetSpeed() {
    return this.speed;
}

// Apply the fuzzified values received from the server
public void ApplyAllSpawnableProperties(string value) {
    // Parse the value and set the pulse rate
    string[] linesInValue = value.Split (',');

    SetColour (getCorrespondingColour(linesInValue [0]));
    SetShapeSize (float.Parse(linesInValue [1]));
    fauxPulseRateTemp =  (float.Parse(linesInValue [2]));
    SetSpeed (float.Parse(linesInValue [3]));

    // Make the Spawnable Pulse
    InvokeRepeating("ModulateSize", 0, fauxPulseRateTemp);
}

// Tween the size of the shape to match pulse rate
void ModulateSize() {
    ScaleTween();
}

// Move towards user
```

```
    void ToggleSingularity() {
        runSingularity = true;
    }

    // Get the actual colour by mapping the name to colour values as can be seen in the source above the previous method.
    Color getCorrespondingColour(string mapper) {
        switch (mapper) {
        case "ifr":
            return new Color(54.0f/255.0f, 0f, 0f);
        case "deepred":
            return new Color(172.0f/255.0f, 0f, 0f);
        case "red":
            return new Color(255.0f/255.0f, 0f, 0f);
        case "redorange":
            return new Color(255.0f/255.0f, 51.0f/255.0f, 0f);
        case "orange":
            return new Color(255.0f/255.0f, 102.0f/255.0f, 0f);
        case "yellow":
            return new Color(243.0f/255.0f, 255.0f/255.0f, 0f);
        case "lemon":
            return new Color(181.0f/255.0f, 255.0f/255.0f, 0f);
        case "green":
            return new Color(40.0f/255.0f, 255.0f/255.0f, 0f);
        case "cyan":
            return new Color(0f, 255.0f/255.0f, 229.0f/255.0f);
        case "blue":
            return new Color(0f, 122.0f/255.0f, 255.0f/255.0f);
        case "blueind":
            return new Color(5.0f/255.0f, 0f, 255.0f/255.0f);
        case "violet":
            return new Color(51.0f/255.0f, 0f, 51.0f/255.0f);
        case "deepviolet":
            return new Color(58.0f/255.0f, 0f, 85.0f/255.0f);
        case "nearuv":
            return new Color(34.0f/255.0f, 0f, 34.0f/255.0f);
        }

        return new Color(0f, 0f, 0f);
    }

}
```

Background and Collider Colour Fader

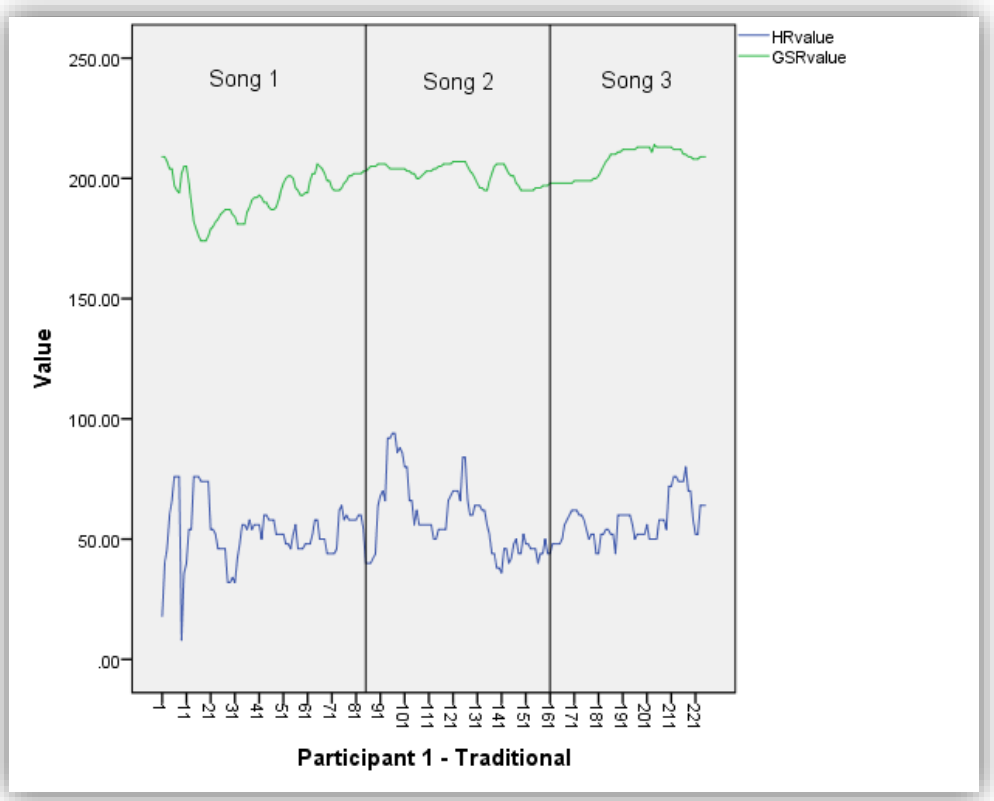
List the code from the 'ThesisCamera.cs' script.

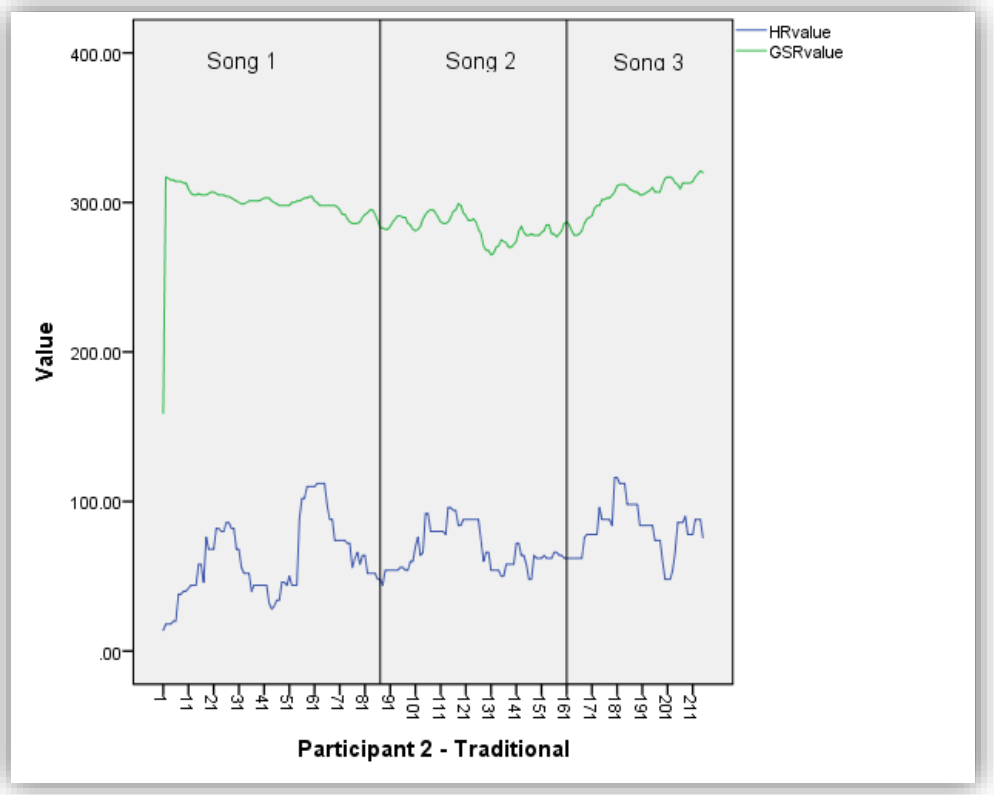
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using UnityEngine;
using UnityEngine.UI;
using DG.Tweening;

/// <summary>
/// Thesis camera used for Background colour fading and white flash
/// </summary>
namespace Assets
{
   class ThesisCamera : MonoBehaviour
   {

      private Camera _cameraMain;
      public int _colorCount = 3;
      private System.Random random;
      private Color currentColor;
      private int index = 0;
      int nextIndex;
      public List<Color> colorList;
      public bool flash;
      public float speed = 1f;
      float startTime = 0f;
      float progress = 0f;
      float flashProgress = 0f;


      private void Start()
      {
         random = new System.Random();

         var mainCamera = GameObject.FindGameObjectWithTag("MainCamera");
         _cameraMain = mainCamera.GetComponent<Camera>();
         _cameraMain.clearFlags = CameraClearFlags.SolidColor;
         _cameraMain.backgroundColor = Color.red;
         Debug.Log(_cameraMain.fieldOfView);
         colorList = new List<Color>();

         // For every colour
         foreach (int i in Enumerable.Range(0, _colorCount))
         {
            colorList.Add(new  Color(random.Next(0,  75)  /  255.0f,  random.Next(0,  75)  /  255.0f,
random.Next(0, 75) / 255.0f));
         }

         currentColor = colorList[0];
         nextIndex = (index + 1) % colorList.Count;
      }

      private void Update()
      {
```

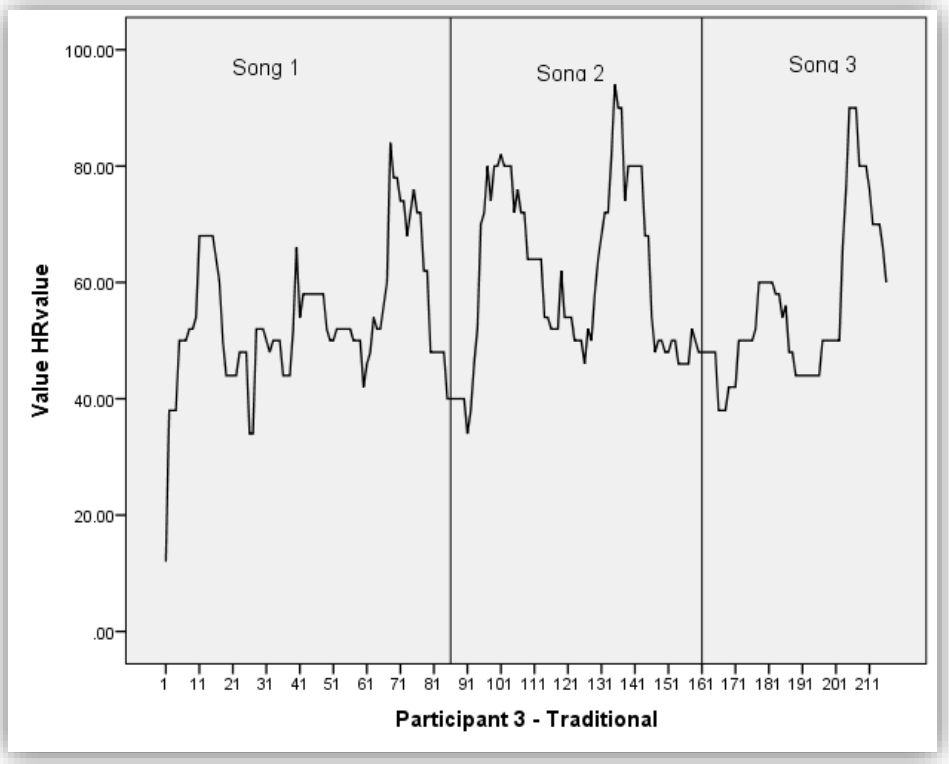
```
        // If box did not hit user
        if (!flash)
        {
            // Keep moving towards user
            progress = (Time.time - startTime)/speed;
            if (progress >= 1)
            {
                nextIndex = (index + 2)%colorList.Count;
                index = (index + 1)%colorList.Count;
                startTime = Time.time;
            }
            else
            {
                currentColor = Color.Lerp(colorList[index], colorList[nextIndex], progress);
            }

            _cameraMain.backgroundColor = currentColor;
        }
        // if hit - flash white screen
        else{
            if (flashProgress < 1)
            {
                flashProgress += Time.deltaTime/3;
                _cameraMain.backgroundColor = Color.Lerp(Color.white, colorList[index],
flashProgress);
            }
            else
            {
                flashProgress = 0;
                flash = false;
            }
        }
    }

    public void Flash()
    {
        flash = true;
    }

  }
}
```

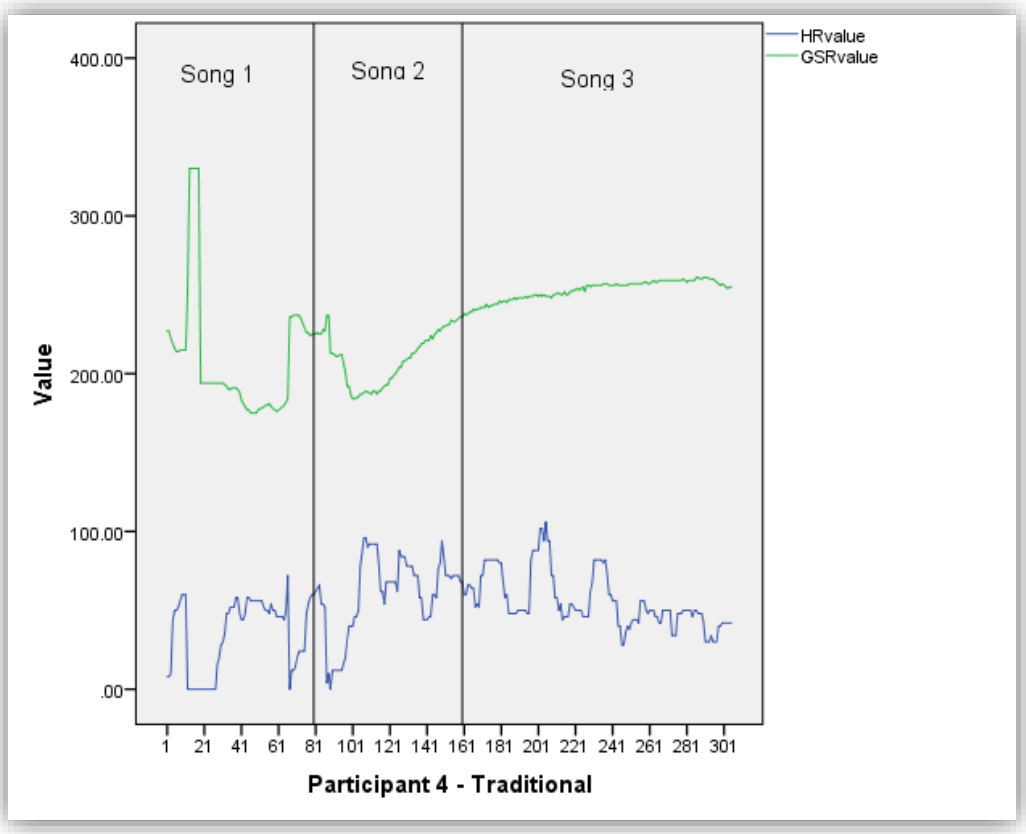
12.2 Section 2: Line charts for Traditional participants

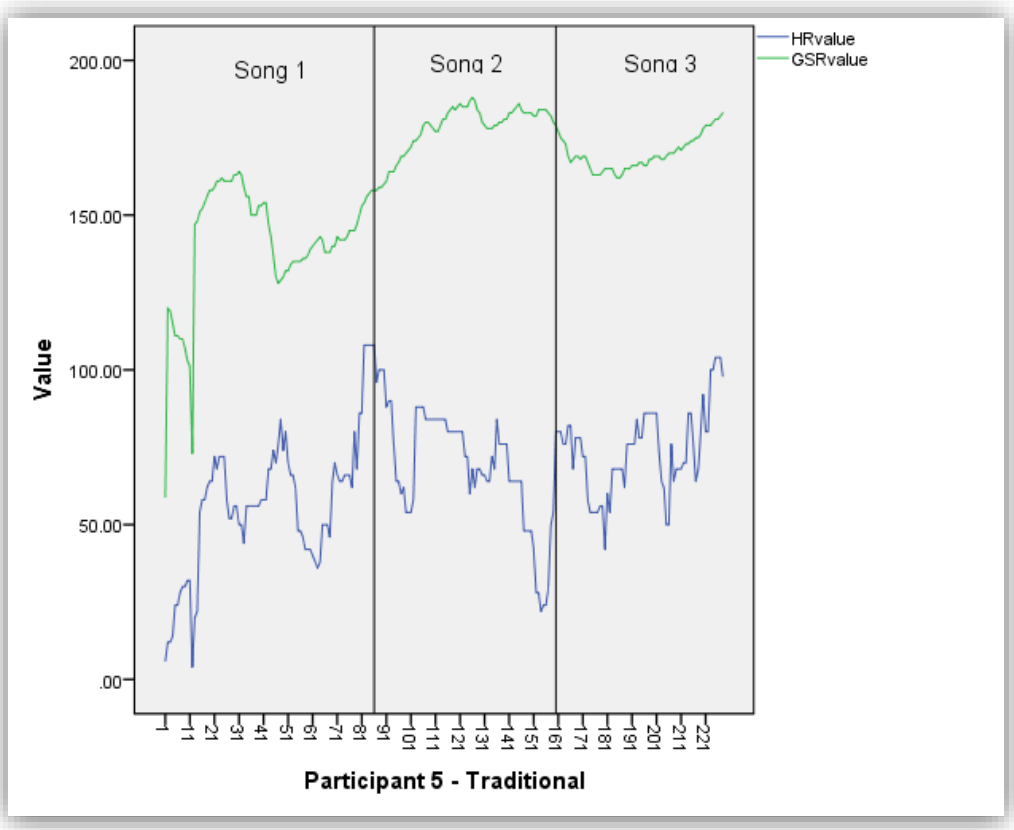
Traditional

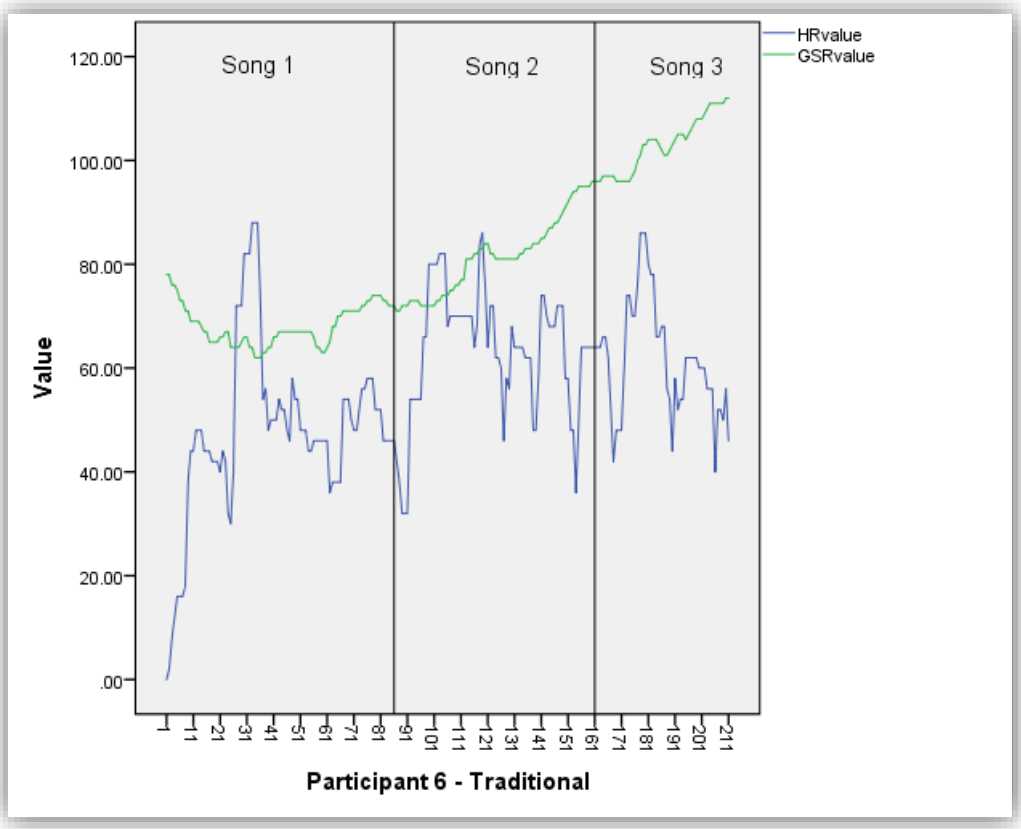


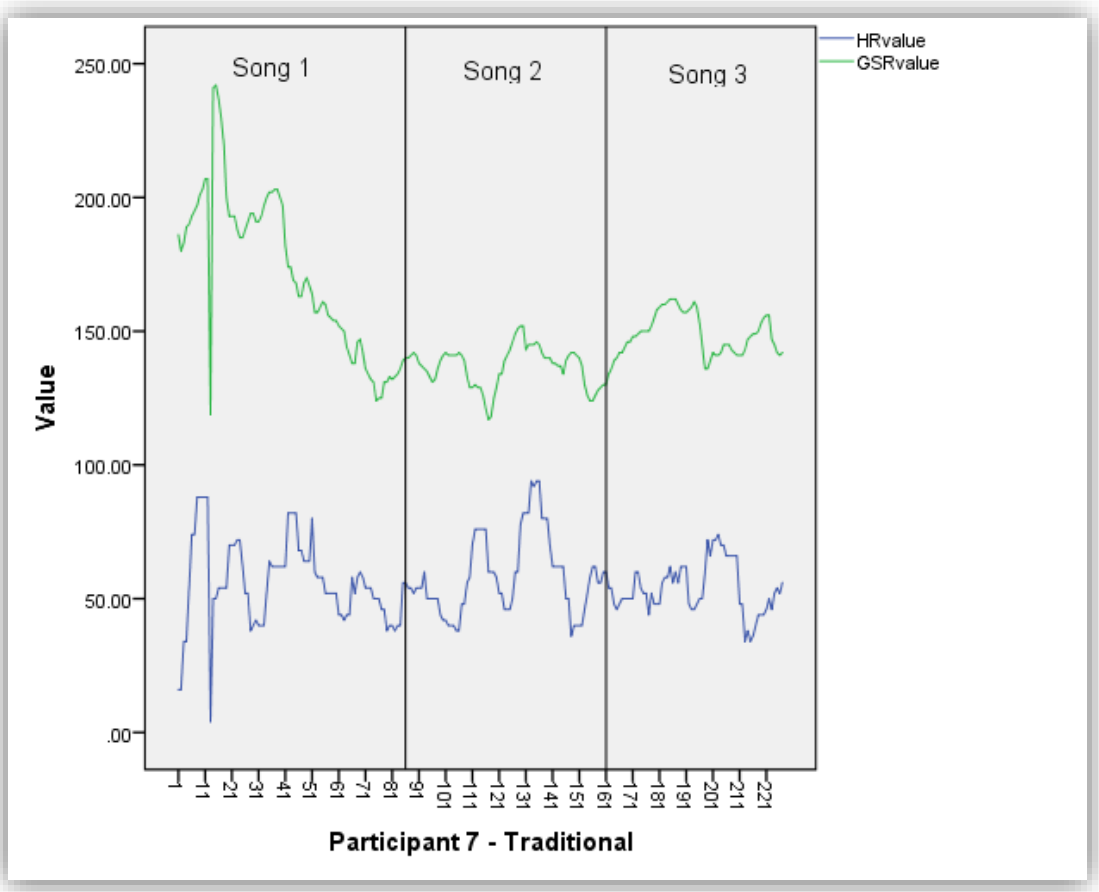


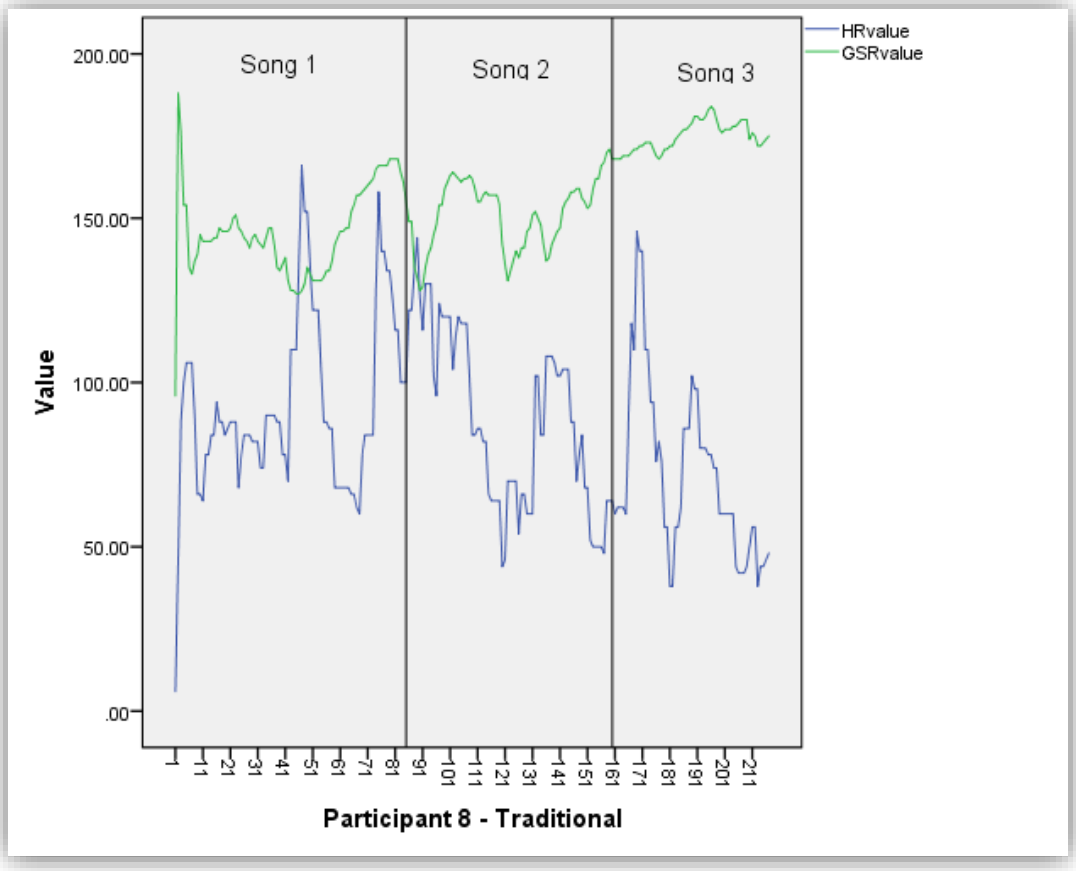

Song 1
Song 2
Song 3
Value HRvalue
100.00
80.00
60.00
40.00
20.00
.00
1 11 21 31 41 51 61 71 81 91 101 111 121 131 141 151 161 171 181 191 201 211
Participant 3 - Traditional

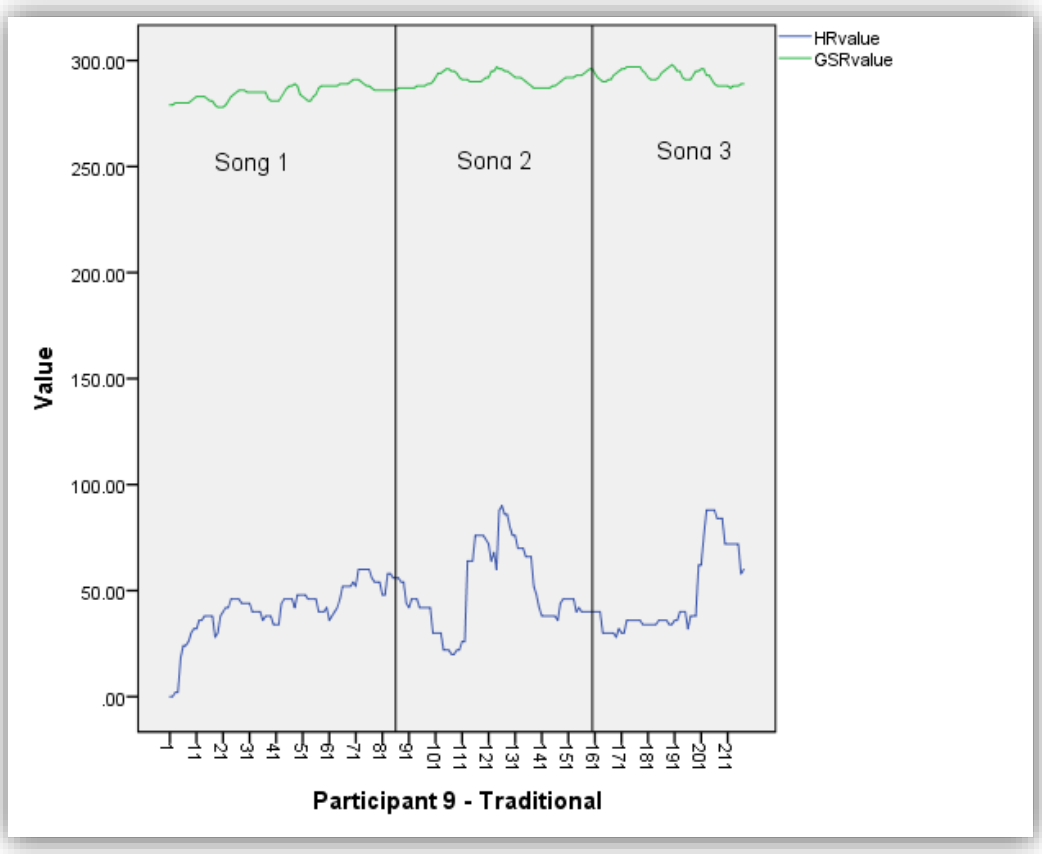

HRvalue
GSRvalue
Song 1
Song 2
Song 3
Value
400.00
300.00
200.00
100.00
.00
1 21 41 61 81 101 121 141 161 181 201 221 241 261 281 301
Participant 4 - Traditional

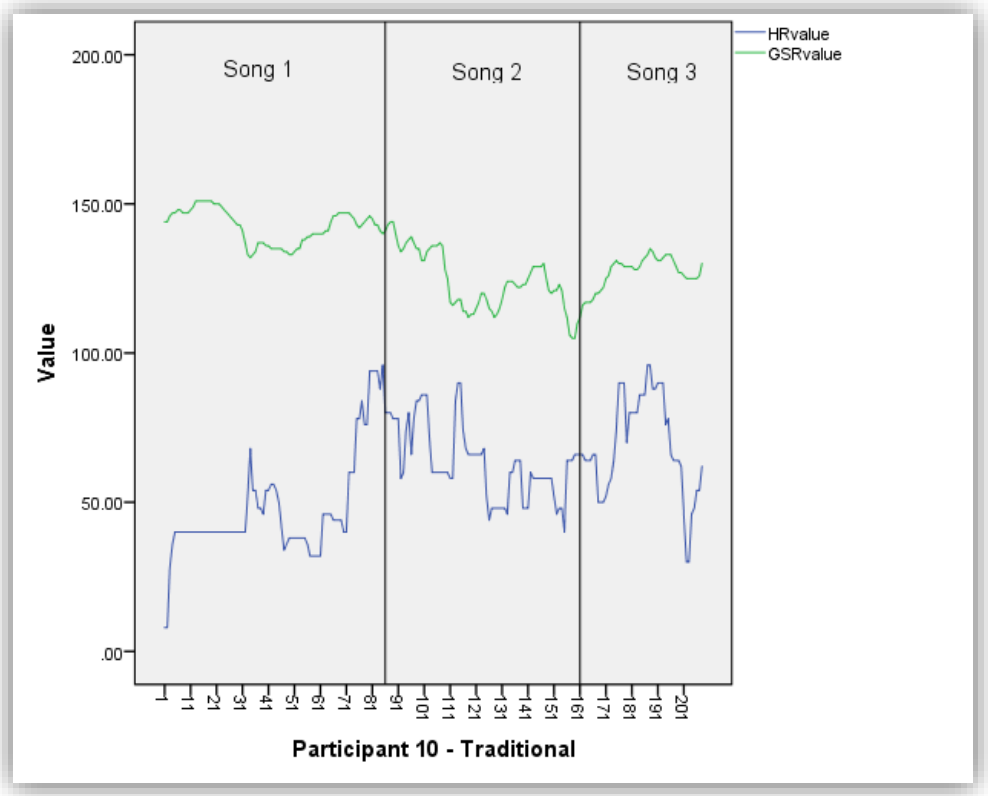

Song 1
Song 2
Song 3
HRvalue
GSRvalue
Value
200.00
150.00
100.00
50.00
.00
Participant 5 - Traditional


Song 1
Song 2
Song 3
HRvalue
GSRvalue
Value
120.00
100.00
80.00
60.00
40.00
20.00
.00
Participant 6 - Traditional


Song 1
Song 2
Song 3
HRvalue
GSRvalue
Value
250.00
200.00
150.00
100.00
50.00
.00
Participant 7 - Traditional


Song 1
Song 2
Song 3
HRvalue
GSRvalue
Value
200.00
150.00
100.00
50.00
.00
Participant 8 - Traditional


HRvalue
GSRvalue
Song 1
Song 2
Song 3
Value
300.00
250.00
200.00
150.00
100.00
50.00
.00
Participant 9 - Traditional


HRvalue
GSRvalue
Song 1
Song 2
Song 3
Value
200.00
150.00
100.00
50.00
.00
Participant 10 - Traditional

12.3 SECTION 3: RESULTS JARGON

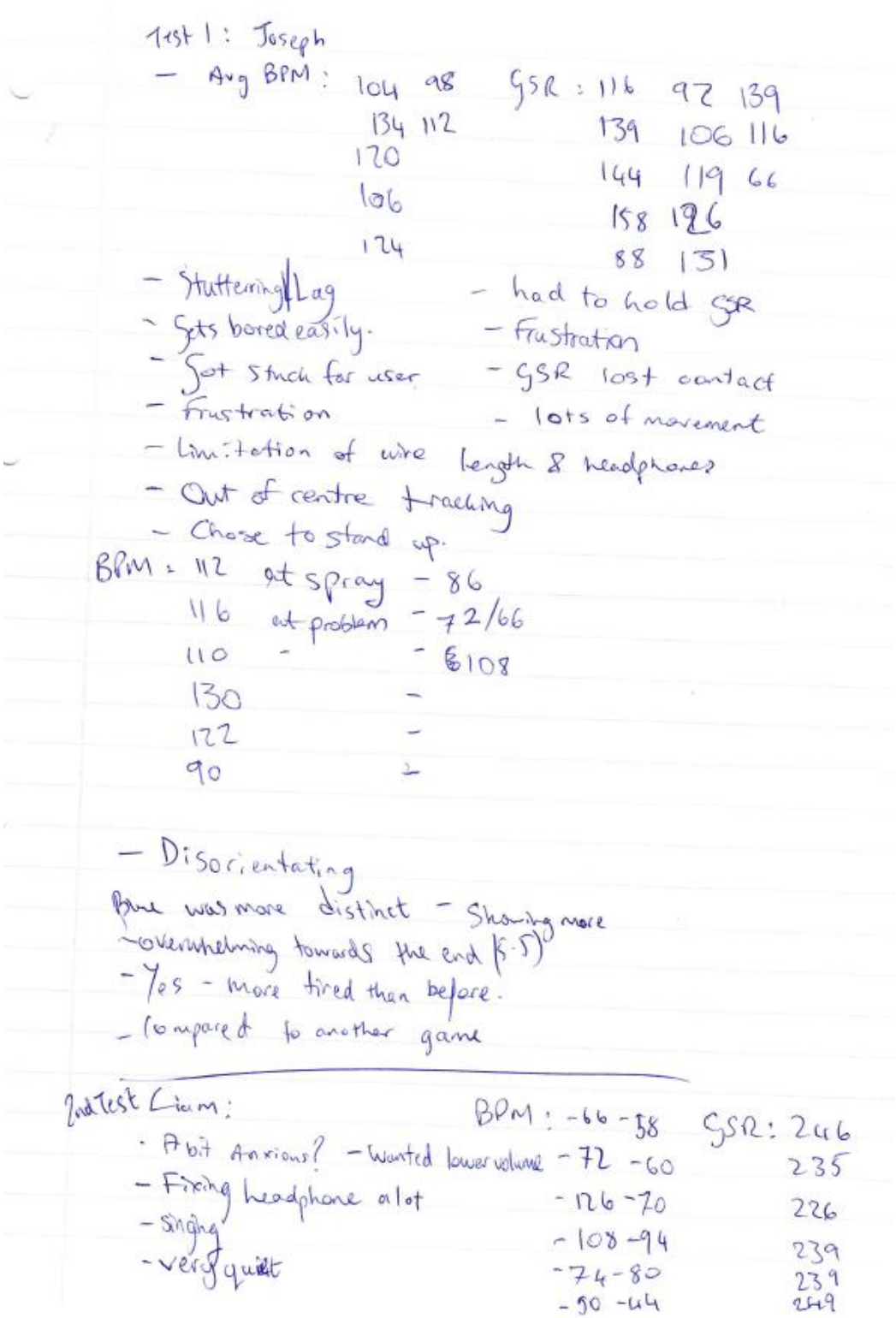
12.3.1 **Principal Component Analysis Data sets**

Participant_Number	Difficulty	Duration	Overwhelming	Restricted	Restless	PCA_X	PCA_Y
VR_Participant_1	2	4	6	2	5	1.82889	0.49846
VR_Participant_2	4	3	1	1	2	-0.43794	-1.61168
VR_Participant_3	3	2	3	2	1	-1.04855	-0.67573
VR_Participant_4	4	3	1	2	1	-0.83128	-1.08235
VR_Participant_5	2	3	5	4	6	0.78276	1.19627
VR_Participant_6	3	3	1	3	6	0.25366	-0.49336
VR_Participant_7	2	4	4	1	5	1.82067	-0.65963
VR_Participant_8	2	3	2	2	6	0.87003	-0.81674
VR_Participant_9	2	3	1	2	1	-0.20436	-1.23197
VR_Participant_10	5	3	1	4	1	-1.54314	0.09595
Traditional_Participant_1	5	3	2	4	1	-1.43942	0.39913
Traditional_Participant_2	2	3	1	2	1	-0.20436	-1.23197
Traditional_Participant_3	3	4	4	2	3	0.91974	-0.0779
Traditional_Participant_4	3	4	5	2	3	1.02345	0.22528
Traditional_Participant_5	5	3	7	4	1	-0.92087	1.915
Traditional_Participant_6	2	3	2	3	4	0.28255	-0.30982

Traditional_Participant_7	2	3	5	4	1	-0.18792	1.08421
Traditional_Participant_8	2	3	5	4	4	0.39449	1.15145
Traditional_Participant_9	3	3	6	4	2	-0.20353	1.48461
Traditional_Participant_10	5	3	1	4	3	-1.15486	0.14078

12.3.2 **Notes taken during experiment observation for each participant**

12.3.2.1 VR Participants



Test 1: Joseph
- Avg BPM: 104 98 134 112 120 106 124
GSR: 116 92 139 139 106 116 144 119 66 158 126 88 131
- Stuttering/Lag
- Gets bored easily.
- Got stuck for user
- Frustration
- Limitation of wire length & headphones
- Out of centre tracking
- Chose to stand up.
- had to hold GSR
- Frustration
- GSR lost contact
- lots of movement

BPM: 112 at spray - 86
116 at problem - 72/66
110 - - 108
130 -
122 -
90 -

- Disorientating
Pain was more distinct - Showing more overwhelming towards the end (5.5)
- Yes - more tired than before.
- Compared to another game

2nd Test Liam:
- A bit Anxious? - Wanted lower volume
- Fixing headphone a lot
- singing
- very quiet

BPM: -66 -58 -72 -60 -126 -70 -108 -94 -74 -80 -90 -44
GSR: 246 235 226 239 239 249

Liam - cont - headphone was an issue;

BPM	GSR
48	250
44	251
66	251
110	102
106	
82	

- Didn't know what it was - They look similar
- was a destruction game
  - found a strategy
  - Take the big box approach
- Music vital to the experiment
  - without good music - would have failed.
- Cool experiment

Kris - tipsy & stoned

BPM	GSR
102 - 92	241 - 279
- 90 - 94 - 72	255 - 259
78 - 80	247 - 248
74 - 56	245
50 - 44	243
72 - 68	281
86 - 82	294
96 - 54	284

- Questioned what to do
- Questioned why it lags
  - Stopped moving
- Looks immersed
- concious of wires - wire restriction (for vis. destroy)
- Orientation
- They look similar
- Laughed at similarity
- Up with instincts after experiment: due to randomness & instinct
  - get it out of the way
  - Enjoyed all tunes: liked sound - otherwise no.
- Delete by beat
- Touch/feel & Smell/Fragrance - Less Tired due to experimen
- Lag 1-5 : 2/3 (bad is 5) - more alert

Samuel - Test 4: 72 - 75

GSR: 100, 110, 110, 111 - 113, 121, 126, 125, 120, 118, 105, 120 | 116, 112, 124, 122, 139/140, 133, 138, 135, 134

BPM: 46, 52, 42, 58, 64 - 68, 64, 84, 84, 78, 44 | 46, 34, 56, 92, 86, 66, 80, 69, 46, 92, 38 | ~~78~~

- Why would you terminate
  - thinking loop - negative thoughts
  - Rest.
- More Alert now
  - less tired.
- Developed mini games

- very calm
- 1 hand down
- didn't destroy boxes
- Lifted hand 122 - 92 (GSR HR)
- Learnt how to destroy 105 - 34
- Time was good
- immersion was required
- Close to good

Kris C: 71, 167

GSR: 120, 125, 140, 147, 142, 139, 154 | 141, 136, 125, 115, 108, 114, 111, 107 | 120, 109, 117, 112

BPM: 64, 78, 90, 62, 50, 90, 86 | 84, 64, 58, 68, 70, 66, 92 | 104, 110, 88, 42, 72, 70, 64 | 64

- Bit fidgity, wires pulled more
- Continuously moving to kill boxes
- Smiled at smell of spray
- Very Alert
- 110 BPM - went fast at Button
- Adjusted to rag
- wanted to know what stage are & forgot to ask.

- was fun
- Arm got ~~tired~~ & stopped shooting
- A little more tired.
- Restrictive colour palette.
- Found a relativity with speed to GSR & HR to tween

Ashley: Test 6: BPM: 64, GSR: 148

BPM:				GSR:			
60	38/46	60	~~80~~	147	159	181	165
58	40	52	72		162	183	177
56	34	54		151	166	186	161
60/76	68	48		149	166	154	161
60	74	48		143	168	154	164
~~70~~	50	62		115	175	163	
36	50	72		148	177	165	
				175	179		

[redacted]

Comments:

- Very quiet
- Questioned rotation
- Not much movement
- Put hand down mid second song - SR went down By 20
- Put back up 10 seconds later
- Not shooting much.
- More quiet
- Feel more Awake
- No Disorientation.
- Created your own pattern - want for further analyses
- Pattern Match - Color killing

Luke Lanzon - Test 7: BPM: 56 GSR: 226

GSR: 226	253	274	BPM: 36	68	56
~~228~~ 240	265	271	72	62	70
235	267	264	56	66	68
242	251	264	74	96	50
243	269	255	66	84	66
238	270		64	40	82
232	271		74	78	
247	267		44	52	
244	269		62	60	

Comments:

- Enjoying Music
- Humming 'Wow'
- Grumbling About Movement
- Had a fright to notification
- Coughed at spray
- Struggled by rotation - Stood up
- Much more immersed when standing
- Singing
- Screen rotated too much
- Saw a relation when feeling stressed
- Very tired from Arm
- Wires interfered
- Rotated A lot
- Grumbled about being tired
- Saw relation of speed with state

Craig - Test 8 : GSR : 176 BPM: 75

GSR: 147	~~171~~	184	168
138	173	183	173
140	168	185	170
143	170	183	175
145	151	151	166
161	~~80~~	169	
	180	177	
		165	

BPM: 60	66	118	76
62	56	92	84
94	52	64	80
102	64	140	92
118	42	124	96
84	72	98	
72	90	102	
	84	96	
		72	

Comments:

- Very quiet
- Never said a word
- no connection of movement to HR
- Pretty normal speed approach
- Sweating & breathing
- Pans to the right a lot - Frustrating
- From end of last song
- more tired now
- At times the pulsing went quite fast.
- worried about wrong things

Luke G - Test 9 : GSR: 130 BPM: 82

GSR:			BPM:		
123	109	127	66	110	76
99	109	126	72	96	95
98	116	120	74	102	118
93	115	117	100	110	108
93	76	114	120	120	78
115	84		116	86	
105	101		66	64	
99	122		102	106	

Comments:
- Panning of screen
- Appreciated Queen
- Requested reason for Flag>
- Constantly killing
- Queen Stopped - Remarked
- Boxes Approached me slowly.
  - All come at once
- Quite spaced out not erratic
- Was restricted by movement in wires
- Unaware of seating

Raymond - Test 10 : GSR: 276 BPM: 84

- No sensory overload

GSR:				BPM:			
273	263	275	266	72	112	62	142
265	270	281	276	58	114	92	52
265	265	285	279	78	78	180	72
267	268	269	280	112	62	112	44
268	270	259	291	116	110	82	50
262	262	256		88	124	98	
261	273	259		900	82	96	

Comments:

- Singing
- Enjoying the music & Boxes
- Started destroying boxes
- Wow Remarks
- Enjoying Singing
- Smiling & content/Jiggling
- levels rose with climax of song
- Heavy Breathing
- Camera Panned 62 BPM
- Stationary
- Very Quiet
- Dropped HR at Techno

- Did you feel Anxious/Nervous?
  - Same state in and out
- Did you feel irritated at Robots or Bored
  - no
- Did you feel a resemblence with speed & HR?
  - nice visualisation, no relation to what was felt.
- Felt like it was short.

12.3.2.2 Traditional Participants:

(Traditional) ZAN – Test 1: GSR: 251 208 BPM: 81 84

GSR: BPM:

Comments:
- Quiet
- Killing a lot of boxes
- Going Fast
- Slowed down At Points
- Got faster with Climax & The Queens Song
- Stopped killing
- Clearing up Boxes to see the Scene
- Clearing spree
- Got Relaxed towards end of Jackson.
- Waited for Boss to come up

Questionnaire 1
- Bit less fast – more correct.
- Easier
- Overwhelmed by overload of Zombies
- Slightly ~~less~~ tired → more active now
- feels more engaged

Visual:
- When sped up – felt engaged
- Instant reaction
- Not Anxious or Irritated → except clearing the screen

Colours:
- White Joint Strike
  - overpowering

Music:
- escalation of transition → experience transition
- Not distracted

Daniela - Test 2 : GSR: 302 320 BPM: 66 60

Comments;
- More time spent answering Quiz
- Singing & Destroying → Enjoying it
- Perplexed with Music - playing with pattern
- Slowed down as Queen slowed/meandered
- Happy with Song 2.
- Communicating with Experimentor.
- Couldn't keep up with Boxes
- Moved to the Beat of Tiga
- Got tired at Tiga (Song 3)
- Overload of Boxes
- Looked Tired
- Never stopped clicking
- Enjoying Destruction

Questionnaire :
- Slowed Down even more in Second Quiz

Engagement;
- wanted to go on.
- keep destroying Boxes

Visual:
- Visual got Faster/Couldn't keep up.
- Didn't think about anything else other than destroying boxes/Game/Fun

Music:
- Enjoyed every song even Tiga
- More immersed with Queen

Irritability/Nervous/Anxious:
- Never

Overall:
- Less tired
- Music Woke me up
- Feels Alive

In a sentence :
- ~~How~~ Feels like Fun
- Want to learn.
- More positive than any negatives

Dylan Bugattil - Test 3: GSR: 64 BPM: 63
→ Lack of Conductivity

Comments:
- looks excited
- issues with GSR
- Didn't start destroying
- Asked if sprayed
- started destroying
- Communicated with Experimentor.
- Really likes the Music
- Very Relaxed State
- Feels the need to inform about state
- Sounded a b.t bored/agitated
- Informed transition of third song
- Realised smell instantly - Highly communicative
- Yawned in third Song.

Qurstionaire":
- Easier Flags - knew them
- Restless cause disliked Tiger Sound
- Overwhelming:
  - Too many at once
  - No chance to know representation of boxes
- Lost in the App

Anxious for the next song:
- Expected the same genre
- Music had familiar tempo

Not Engaged - Just to visualise
- Relation with pulse, speed was shown - wanted to rid of 3rd Song

Overall:
- Insight on how songs affect our state - Positive Experience
- it excited become looking forward
- the not excited - looking forward for it to end.
- Third song made you bored/Anxious → Background Sound.
- Take consideration How long questionnaire took to answer.

Manuel – Test 4 : GSR: 247 BPM: 75

Comments:
- Lots of issues with Internet & the upload
- Instant killing.
- Working in Patterns
- Very Focused
- Didn't move an inch
- Clearedscreen
- seemed relaxed
- Once again cleared screen
- no communication -
- slowed down mid-last song.
- No slow down on Questionnaire

Questionnaire:

Any Anxiety/Irritability?
(Headache'ish) - you felt more tired at the end
- Yes, combining sensors, no movement & overwhelming

Any Relation to Speed & Pulse
- Yes, in fact I felt a bit overwhelmed & believe that there was relation.
- Everything caused problems

Where you bored?
- except for when to stop, NO!

Did you feel engaged?
- Quite engaged/non distracting

Was it interesting?
- If I knew the mappings, it would be better

Overall in one sentence?
- Know the details is important for the feeling to be better
- less stressed if I have more.

Were you sensitive to the colors/light (at the end?)
- White light shocked me

Music?
- Liked: Queen & Tiga - Beat to pieces
- Disliked: Michael Jackson

→ have a bit of a headache

Jurgen: GSR: 111 BPM: 87

Comments:
- Quite fast on Flags
- Viewing only
- Heavy Breathing.
- Choosing not to destroy Boxes
- Tapping Foot
- very Populated
- Third Song, not one box killed
- Seems to like Tiga + tapping foot.
- wanted to click boxes to see flash
- Didn't destroy boxes
- Boxes were very annoying.
- Too many colours & boxes at once
- More Alert than before

Questionaire:

Any Anxiety/Irritation/Distraction
- Distraction with first smell
- Irritated by the boxes - wanted to minimize screen (Quit)

Music?
Disliked 1, 2, liked 3.

Any relation of Pulse & Speed?
- Yes - since he saw a relation in pulse rate. Not in speed

Did you feel engaged?
Yes - with the music & on the lookout for some for result - then saw box pulse

Was it interesting?
- very interesting, specifically to find out results

Overall in one sentence?
- Heart to Pulse Mapping
- very overwhelming to correlate the concept

Why did you choose to leave them by you?
- Inclined to destroy them
- Did not compel to do it.
- wanted to see if something happened

Flags were harder - no pulse = annoying
→ not familiar.

→ Did not know why the boxes where there
→ White light was soothing instead of many boxes ✓ interesting

Maria Test 6: GSR: 75 BPM: 80

Comments:
- Started destroying instantly.
- Faring well in speed very relaxed.
- Lots of miss clicks in attempt to rush.
- slowed down on second song.
- Got even slower towards end.
- looks a bit more tired.
- less time to complete Quiz 2

Questionnaire:

Any Anxiety/Irritation/Distraction?
- Aimed to kill: too fast to kill

Music?
- enjoyed every song

Any relation of pulse & speed?
- Yes, like when trying to kill
- was becoming faster / slowed down when stop.

Did you feel engaged?
- Music - no good music makes less engagement

Was it interesting?
- Music, mapping between feeling & sight
- wanted to know more

Overall in one sentence?
- unconvenient for GSR
- The game aspect was good.
- ~~rest~~ saw a reflection of size to music
- enjoyed the smells
- not overwhelming.

Colour Blind Jonathan B: GSR: 724 BPM: 75

Comments:

- knowledge of flags is less strong
- Seeing the scene
- looked irritated by clicks
- hiding those closest
- Foot moving to music
- Head moving to music
- Nearly cleared screen
- moved foot to M-J song
- miss-clicking
- Same position, never spoke once initial
- Taking longer to decide flags

Questionnaire:

Where you destroying By Colour?
- By perceived danger/proximity.

Any Anxiety/Irritability/Distraction?
- 2 boxes at once
- White colour shock

~~Any~~ Music?
- Enjoyed the same Rhythm

Anything that bothered you?
- Coming from same angle
- perceived more clicks - more boxes

Interesting?
- Took advantage of S.O to overcome the task.

Any Relation of Pulse & speed
- Not perceived

No attention to colour of boxes
- Cubes with rhythm
- less with colours.
- inclined to check the time
- uses the spray at home
- the more you click, the more they come.
- Music helped clear screen
- Clear by pattern.

Birthday in a month - 35

Silvio: GSR: 157 BPM: 102

Comments:

- ~~Questioned speeds~~ what to do.
- Got quite chaotic - lots of Boxes
- no tapping - same pose: got faster
- Asked to see Flag results
- stopped clicks for a bit
- Screen overloaded again.
- last song got chaotic
- lack of flag knowledge but quite fast
- Got hesitant at the end.
- Made sure to revise

Questionnaire

Any Anxiety / Irritability / Distraction?

- ~~Anxious~~ Inclined to remove big Boxes.

Relation of pulse & speed?

- second song
  - the more coming
  - Destroying more in the second spawn.

Destruction by colour?

- Big Box (size)

Did you feel engaged?

- Yes, ignore surrounding to continue task at hand.

Music?

- Neutral to every song

Restless:

- Second song yes
- Third - less
- Big Boxes were more intimidating
- Too many boxes to keep up.

Interesting?

- Biased: due to previous Experience
- Interesting since I want to know more about the experiment

Test 9 - Dunstan: GSR : 280 BPM: 60

comments:

- Waited for a lot of Boxes
- no tapping.
- Towards mid/end Distraction
- 2nd song stopped clicking - moving fingers
- Destroying to beat
- Lots of misclicks
- Always allowed overlooked screen
- Still Position
- Looked away once
- no destruction on 3rd song till halfway
- wanted to know flags

Questionnaire: → Contemplates choice → Not asked for flags.
- Took longish for first quiz - Even longer for second.

Any Irritatability / Anxiety?
→ Distraction?
- Not really
- Irritated by boxes - rather see the universe
- starts to rush for your liking

Any relation of speed & pulse?
- Not attentive for

Music?
- First song preferred ✓
- Last song disliked, MJ-04
- related music with state
- Might be indirectly related

Any other thoughts?
- No mention of lag.
- not bothered

Engaged?
- Was quite aware but still engaged

What bothered you? - color/light etc.
- Not really no,

Interesting?
- Found it interesting from tutors perspective

Destruction Pattern?
- Not difficult
- Didn't feel like clicking.
- Trying to destroy to expose area
- Destroy large boxes

18-34 Emily - Test 10 : SSR: 102 BPM: 70

Comments:

- Quiet Hearing Boxes approach
- Still position
- Did not destroy - mid Queen
- Stayed in the same position
- Sighed
- looked visibly bored
- swaying to MJ song.
- Intuitive thought leads at MJ.
- Never felt inclined to destroy.
- Visualisation Crashed.
- Very fast at questionnaire
- got cold hand from not moving

Questionnaire:

Anxiety / Irritation / Distraction? -> or Bored?

- thought about fright

Any relation of pulse & Speed?

-No

Music?

- liked all songs
- first most

Overall in one sentence?

Did you feel engaged?

- Quite engaged

If Clicked, what pattern of destroy?

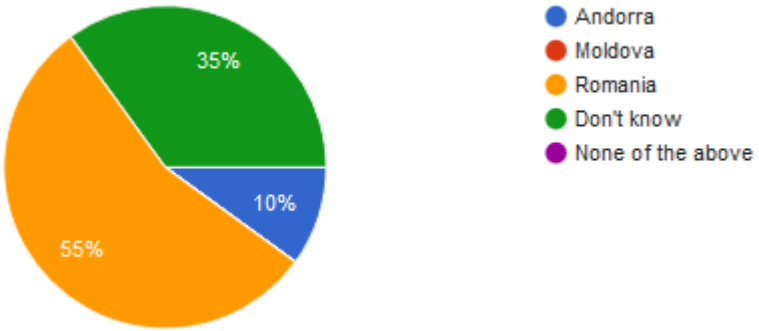
- Never

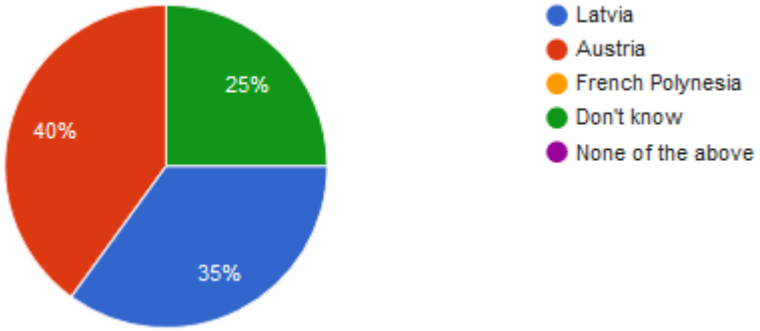
Was it interesting?

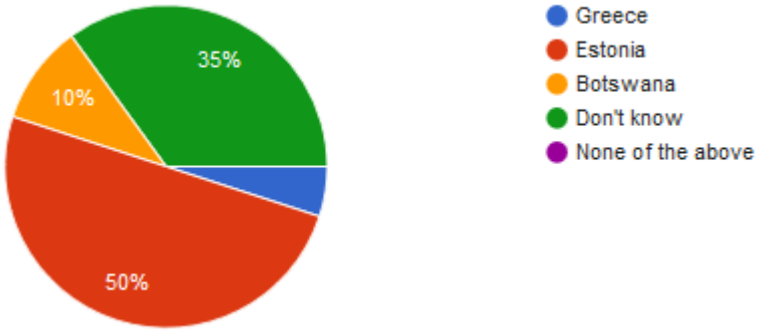
- Trying to see HR with Music
- Interested

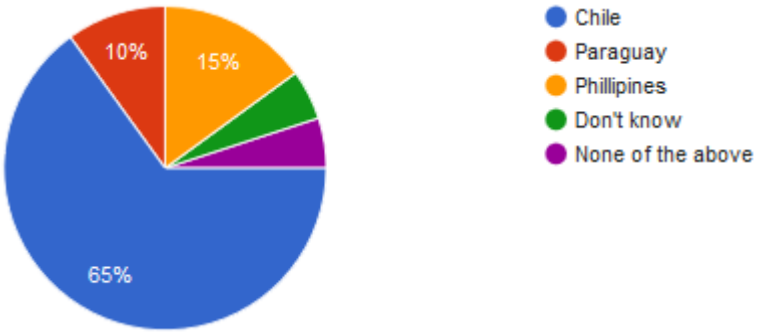
12.3.3 **Questionnaire Cognitive test results**

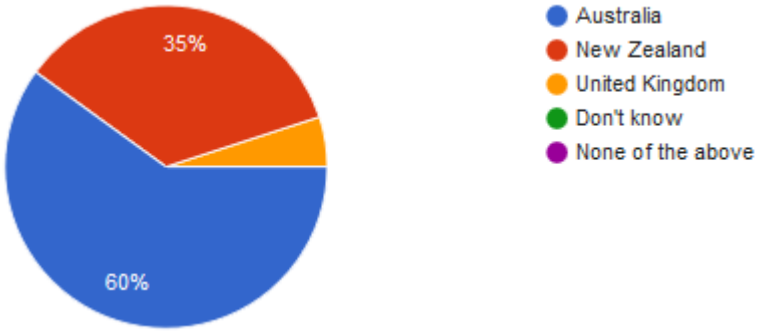
12.3.3.1 Prior-experiment test

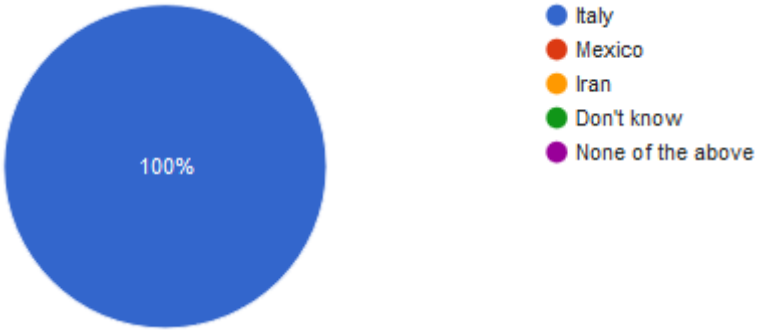


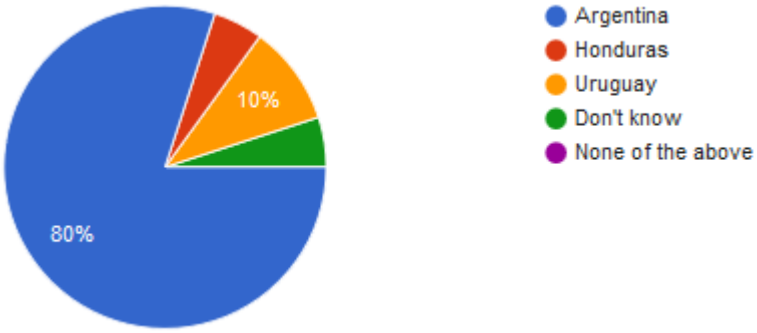


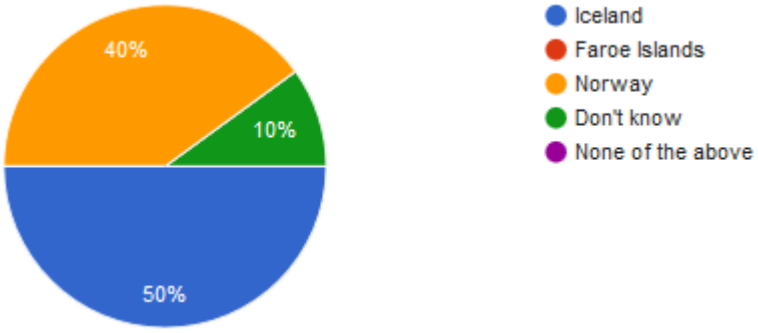


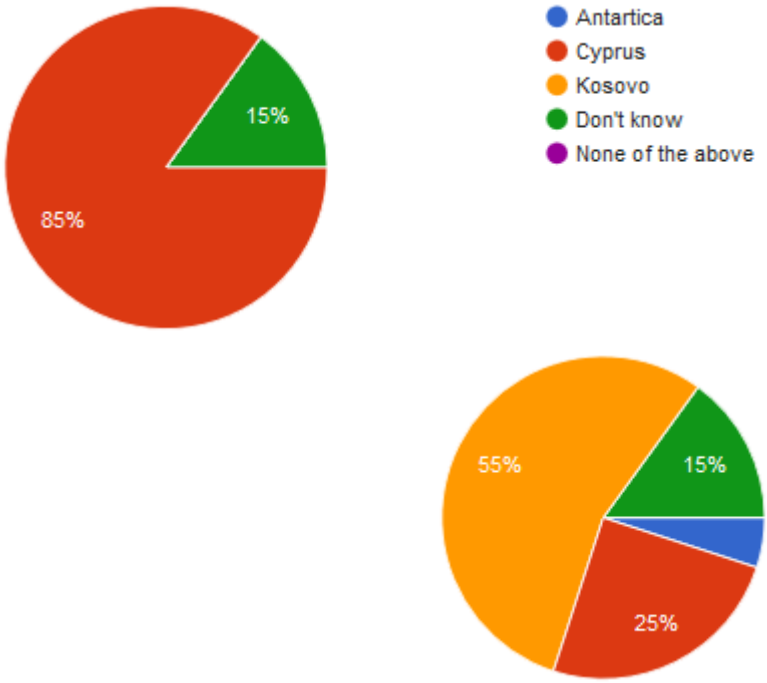



35%
60%
Australia
New Zealand
United Kingdom
Don't know
None of the above

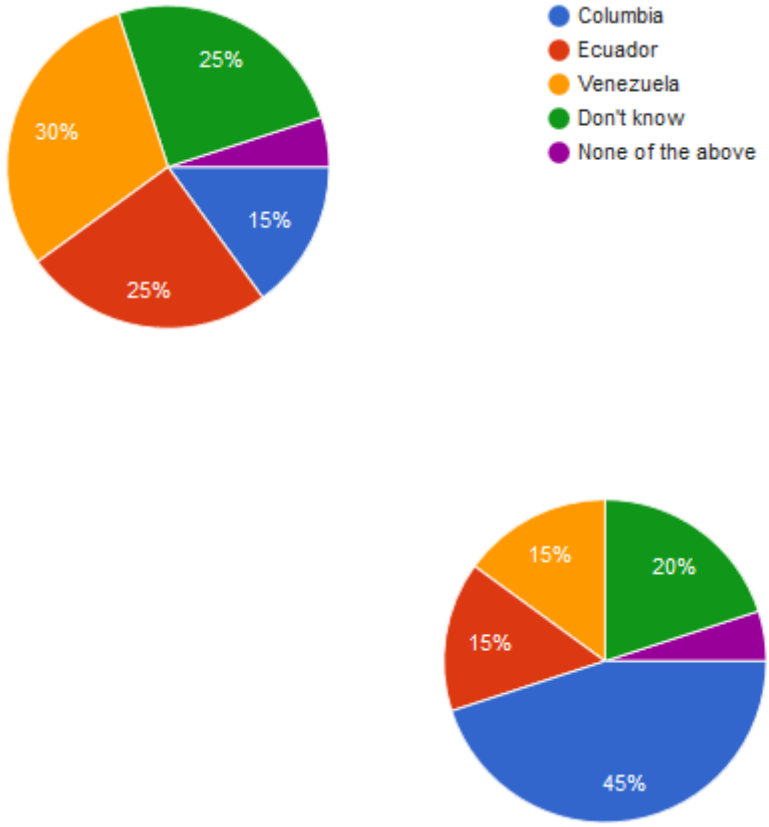

100%
Italy
Mexico
Iran
Don't know
None of the above

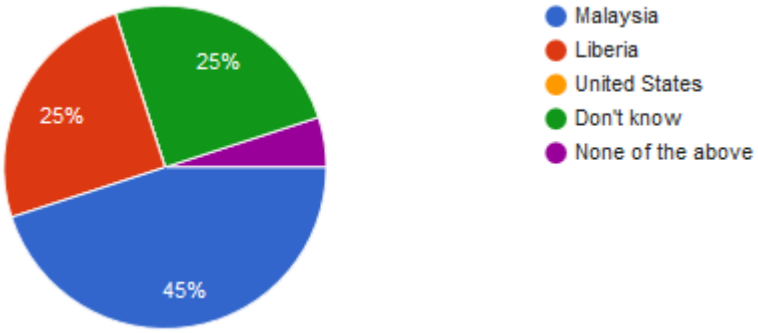

10%
80%
Argentina
Honduras
Uruguay
Don't know
None of the above

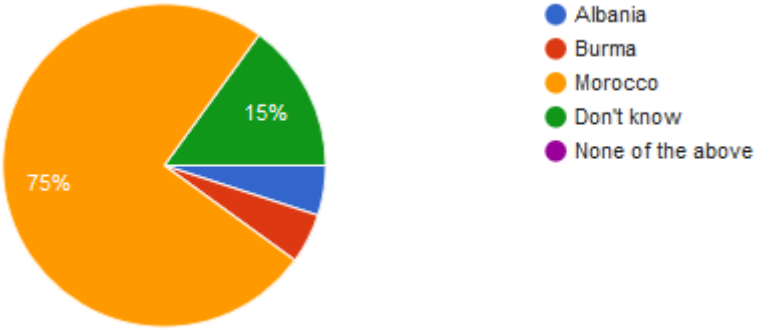

40%
10%
50%
Iceland
Faroe Islands
Norway
Don't know
None of the above

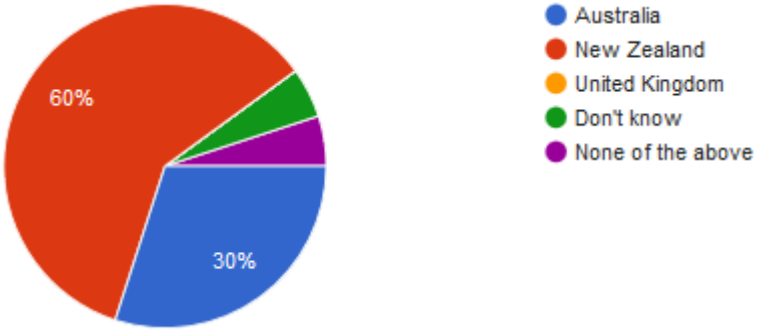


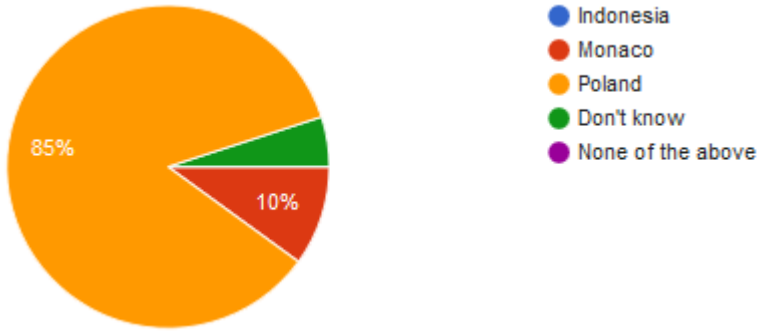
12.3.3.2 Post-experiment test

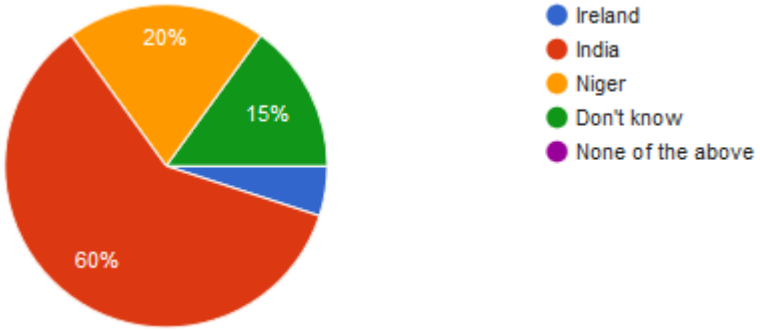


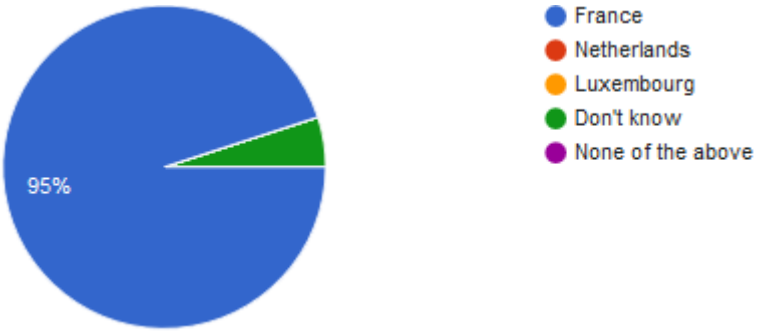

25%
25%
45%
Malaysia
Liberia
United States
Don't know
None of the above

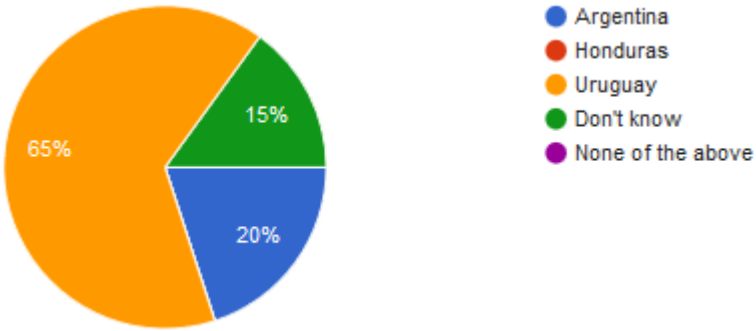

15%
75%
Albania
Burma
Morocco
Don't know
None of the above

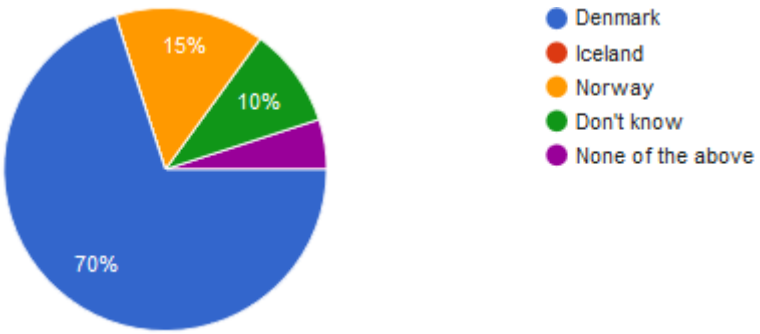

60%
30%
Australia
New Zealand
United Kingdom
Don't know
None of the above


85%
10%
Indonesia
Monaco
Poland
Don't know
None of the above


20%
15%
60%
Ireland
India
Niger
Don't know
None of the above


95%
France
Netherlands
Luxembourg
Don't know
None of the above


15%
65%
20%
Argentina
Honduras
Uruguay
Don't know
None of the above


15%
10%
70%
Denmark
Iceland
Norway
Don't know
None of the above

REFERENCES

000webhost. (2016, March 26). *Free Web Hosting with PHP, MySQL and cPanel, No Ads*. Retrieved from 000webhost: https://www.000webhost.com/

Apache. (2016, March 29). *FTPClient*. Retrieved from commons.apache: https://commons.apache.org/proper/commons-net/apidocs/org/apache/commons/net/ftp/FTPClient.html

Apple, I. (2016, March 19). *OS X Yosemite: Use Dictation Commands to tell your Mac what to do - Apple Support*. Retrieved from support.apple: https://support.apple.com/en-us/HT203085

Arduino. (2015, December 21). *Frequently Asked Questions*. Retrieved from Arduino: https://www.arduino.cc/en/Main/FAQ

Arduino. (2016, April 6). *Download the Arduino Software*. Retrieved from Arduino: https://www.arduino.cc/en/Main/Software

Autism Speaks. (2016, February 20). *Facts about Autism*. Retrieved from Autism Speaks: https://www.autismspeaks.org/what-autism/facts-about-autism

AutismSpeaks. (2016, February 20). *What Is Autism?* Retrieved from Autism Speaks: https://www.autismspeaks.org/what-autism

Barco. (2016, March 24). *Barco Stereoscopic visualization techniques and immersive data analysis to support cancer research at Erasmus MC, the Netherlands*. Retrieved from Barco: http://www.barco.com/en/News/Press-releases/barco-stereoscopic-visualization-techniques-and-immersive-data-analysis-to-support-cancer-research-a.aspx

Bender, T. (2015, November 16). *java - Print "hello world" every X seconds*. Retrieved from Stack Overflow: http://stackoverflow.com/questions/12908412/print-hello-world-every-x-seconds

Boundless. (2015, December 15). *Introducing the Perception Process*. Retrieved from Boundless.com: https://www.boundless.com/psychology/textbooks/boundless-psychology-textbook/sensation-and-perception-5/introduction-to-perception-39/introducing-the-perception-process-167-12702/

Bourke, P. (2016, March 20). *Calculating Stereo Pairs*. Retrieved from paulbourke: http://paulbourke.net/stereographics/stereorender/

Bown, O. (2015, December 5). *The Beads Project - Realtime Audio for Java and Processing*. Retrieved from beads: http://www.beadsproject.net/

Bradford, A. (2016, March 20). *Deforestation: Facts, Causes & Effects*. Retrieved from Live Science: http://www.livescience.com/27692-deforestation.html

Brownlee, M. (2016, March 24). *Virtual Reality: Explained!* Retrieved from YouTube: https://www.youtube.com/watch?v=i4Zt3JZejbg

Burns, M. (2016, March 24). *Today Is The Last Day To Buy Google Glass*. Retrieved from Tech Crunch: http://techcrunch.com/2015/01/19/today-is-the-last-day-to-buy-google-glass/

Campbell, B. (2016, February 22). *Sensory Overload: Sources and Strategies*. Retrieved from CFIDS & Fibromyalgia Self-Help: http://www.cfidsselfhelp.org/library/sensory-overload-sources-and-strategies

Charara, S. (2016, March 26). *Explained: How does VR actually work?* Retrieved from Wearable: http://www.wareable.com/vr/how-does-vr-work-explained

Chastain, S. (2016, March 22). *What is Tagging? Keyword Tags for Photo Organization*. Retrieved from About Tech: http://graphicssoft.about.com/od/glossary/a/tagging.htm

Cherry, K. (2016, March 27). *Cognitive Psychology: The Science of How We Think*. Retrieved from about health: http://psychology.about.com/od/cognitivepsychology/f/cogpsych.htm

chibimaria. (2015, December 15). *What is Information Visualization?* Retrieved from YouTube: https://www.youtube.com/watch?v=vbzozDCW_Bg

Cingolani, P., & Alcalá-Fdez, J. (2015, November 7). *jFuzzyLogic*. Retrieved from jFuzzyLogic.sourceforge: http://jfuzzylogic.sourceforge.net/html/manual.html

Crawley, D. (2016, March 24). *We're not talking about what VR is doing to our eyes … and our brains*. Retrieved from VentureBeat: http://venturebeat.com/2015/04/18/were-not-talking-about-what-vr-is-doing-to-our-eyes-and-our-brains/

dc42. (2016, February 26). *Voltage drop when relay activates and affects analog readings*. Retrieved from forum.arduino: http://forum.arduino.cc/index.php?topic=204714.0

Dewey, D. (2016, April 4). *Measurement and Observer Effects*. Retrieved from intropsych: http://www.intropsych.com/ch01_psychology_and_science/measurement_and_observer_effects.html

Diamond, A. (2016, March 25). *Cognitive Processes in Learning: Types, Definition & Examples*. Retrieved from Study: http://study.com/academy/lesson/cognitive-processes-in-learning-types-definition-examples.html

Dr. Brian Jackson, P. (2016, March 26). *How Does Virtual Reality Work? - Part 1 of 2*. Retrieved from Marxent Labs: http://www.marxentlabs.com/how-does-virtual-reality-work-part-i-of-2/

Feng, H., Mech., C., Technol., I. M., & Huhhot, C. (2016, March 20). *Investigation on human visual response latency*. Retrieved from IEEExplore: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5540888&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D5540888

FileZilla. (2016, March 16). *FileZilla - The free FTP solution*. Retrieved from filezilla-project: https://filezilla-project.org/

Fizzed. (2016, April 5). *RXTX for Java*. Retrieved from Fizzed: http://fizzed.com/oss/rxtx-for-java

Foundation, P. (2016, April 5). *Download | Processing.org*. Retrieved from Processing: https://processing.org/download/?processing

Fritzing. (2016, February 21). *Fritzing Fritzing*. Retrieved from Fritzing: http://fritzing.org/home/

Fustini, D. (2016, February 25). *Pumping Station: One: Ouch! Sensing Galvanic Sk...* Retrieved from element14 Community: https://www.element14.com/community/groups/pumping-station-one/blog/2011/05/08/ouch-sensing-galvanic-skin-response-gsr

Giardini, D. (2016, January 6). *DOTween - Downloads*. Retrieved from Demigiant DOTween [HOTween V2]: http://dotween.demigiant.com/download.php

Google, D. (2016, April 5). *Download and Samples | Cardboard | Google Developers*. Retrieved from Google Developers: https://developers.google.com/cardboard/unity/download#cardboard_sdk_for_unity_and_demo

Harrison, L., & Loui, P. (2016, May 10th). *Thrills, chills, frissons, and skin orgasms: toward an integrative model of transcendent psychophysiological experiences in music*. Retrieved from frontiers in Psychology: http://journal.frontiersin.org/article/10.3389/fpsyg.2014.00790/full

Harvard Business Review. (2016, February 20). *The Science of Sensory Marketing*. Retrieved from Harvard Business Review: https://hbr.org/2015/03/the-science-of-sensory-marketing

hathol. (2016, March 30). *Instantiate GameObject with parameters*. Retrieved from Unity Answers: http://answers.unity3d.com/questions/254003/instantiate-gameobject-with-parameters.html

Hoffman, C. (2016, March 19). *15 Things You Can Do With Cortana on Windows 10*. Retrieved from How To Geek: http://www.howtogeek.com/225458/15-things-you-can-do-with-cortana-on-windows-10/

Humphries, M. (2016, March 26). *Virtual reality has one big unsolved problem: cables*. Retrieved from Geek: http://www.geek.com/games/virtual-reality-has-one-big-unsolved-problem-cables-1638398/

IBM. (2016, March 10). *IBM SPSS Statistics*. Retrieved from IBM: http://www-01.ibm.com/software/analytics/spss/products/statistics/

Ishikawa, A. (2016, March 22). *List of books and articles about Information Retrieval (in Computing)*. Retrieved from Questia: https://www.questia.com/library/science-and-technology/computers-and-the-internet/information-retrieval-computing

jcrada. (2015, November 7). *GitHub - fuzzylite/jfuzzylite: jfuzzylite: a fuzzy control library in Java*. Retrieved from GitHub: https://github.com/fuzzylite/jfuzzylite

jocelynzada. (2015, July 5). *Stress Makes Art: Galvanic Skin Response and Visual Generation*. Retrieved from Instructables:

http://www.instructables.com/id/Stress-Makes-Art-Galvanic-Skin-Response-and-Visual/

Kanuk, L. S. (2016, April 6). *Consumer Behavior, 8th*. Retrieved from abebooks: http://www.abebooks.com/Consumer-Behavior-8th-Schiffman-Leon-Kanuk/1288331313/bd

Kennedy, M. (2016, March 22). *SQL Indexing For Dummies*. Retrieved from Very Noisy: http://www.verynoisy.com/sql-indexing-dummies/

Kenny, G. (2016, March 15). *I Know Everything About You! The Rise of the Intelligent Personal Assistant*. Retrieved from Security Intelligence: https://securityintelligence.com/i-know-everything-about-you-the-rise-of-the-intelligent-personal-assistant/

Knapp, R. B. (2015, December 5). *Fuzzy Sets and Pattern Recognition*. Retrieved from cs.princeton: http://www.cs.princeton.edu/courses/archive/fall07/cos436/HIDDEN/Knapp/fuzzy004.htm

Kumar, & Senthil, A. (2015, December 5). *Fuzzy Expert Systems for Disease Diagnosis - Google Books*. Retrieved from books.google: https://books.google.com.mt/books?id=Ge2fBwAAQBAJ&pg=PA343&lpg=PA343&dq=aggregation+of+consequents+across+the+rules&source=bl&ots=T0rQzB7rjR&sig=PQAtkbkCoy04bKhSsM1ahC3tDBM&hl=en&sa=X&redir_esc=y#v=onepage&q=aggregation%20of%20consequents%20across%20the%2

Leonardo B. (2016, March 26). *LitJSON*. Retrieved from lbv.github: http://lbv.github.io/litjson/

Lieberman, H. (2016, February 22). *Autonomous Interface Agents*. Retrieved from Media Laboratory, Massachusetts Institute of Technology: http://web.media.mit.edu/~lieber/Lieberary/Letizia/AIA/AIA.html

Macy's. (2016, March 6). *What are In-Store Notifications?* Retrieved from the magic of macy's: https://www.customerservice-macys.com/app/answers/detail/a_id/6354/~/what-are-in-store-notifications%3F

Malmlund, M. (2016, March 24). *Top 10 Best Virtual Reality Headsets*. Retrieved from Heavy: http://heavy.com/tech/2015/07/best-vr-virtual-reality-headset-glasses-goggles-oculus-rift-specs-review/

MATLAB, S. (2015, December 5). *What is Mamdani-Type Fuzzy Inference*. Retrieved from MathWorks: http://www.mathworks.com/help/fuzzy/what-is-mamdani-type-fuzzy-inference.html

McLeod, S. (2016, March 24). *Cognitive Psychology*. Retrieved from Simply Psychology: http://www.simplypsychology.org/cognitive.html

Mellon, C. (2016, March 7). *Intelligent Software Agents*. Retrieved from cs.cmu: https://www.cs.cmu.edu/~softagents/multi.html

Memoto, M. K. (2016, March 6). *How Lifelogging is Transforming the Way We Remember, Track Our Lives*. Retrieved from WIRED:

http://www.wired.com/insights/2013/06/how-lifelogging-is-transforming-the-way-we-remember-track-our-lives/

Meyer, R. (2016, February 22). *Even Early Focus Groups Hated Clippy*. Retrieved from The Atlantic: http://www.theatlantic.com/technology/archive/2015/06/clippy-the-microsoft-office-assistant-is-the-patriarchys-fault/396653/

Microsoft. (2016, March 26). *System.Net Namespace*. Retrieved from Microsoft Developer Network: https://msdn.microsoft.com/en-us/library/system.net%28v=vs.110%29.aspx?f=255&MSPPError=-2147217396

Mind Tools, E. (2016, March 24). *Cognitive Load Theory*. Retrieved from Mind Tools: https://www.mindtools.com/pages/article/cognitive-load-theory.htm

Miyao, D. (2016, March 28). *The BlitzWolf VR headset is everything Google Cardboard should have been (Review)*. Retrieved from AndroidGuys: http://www.androidguys.com/2016/03/03/the-blitzwolf-vr-headset-is-everything-google-cardboard-should-have-been-review/

Nielsen, J. (2016, February 21). *10 Usability Heuristics for User Interface Design*. Retrieved from NN/g Nielsen Norman Group: https://www.nngroup.com/articles/ten-usability-heuristics/

Nieuwenhuysen, P. (2016, March 22). *retrieval.pdf*. Retrieved from Vub.ac.be: http://www.vub.ac.be/BIBLIO/nieuwenhuysen/courses/chapters/retrieval.pdf

Oracle. (2016, March 26). *SerialPortEventListener*. Retrieved from docs.oracle: https://docs.oracle.com/cd/E17802_01/products/products/javacomm/reference/api/javax/comm/SerialPortEventListener.html

O'Reilly. (2016, March 24). *Creve Maples*. Retrieved from O'Reilly Community: http://www.oreilly.com/pub/au/4748

org.json. (2016, March 8). *Maven Repository: org.json >> json >> 20140107*. Retrieved from MVNRepository: http://mvnrepository.com/artifact/org.json/json/20140107

Ripton, J. (2016, March 28). *Google Cardboard: everything you need to know*. Retrieved from techradar.: http://www.techradar.com/us/news/phone-and-communications/mobile-phones/google-cardboard-everything-you-need-to-know-1277738

Rouse, M. (2016, March 24). *What is virtual reality?* Retrieved from Tech Target: http://whatis.techtarget.com/definition/virtual-reality

Saul McLeod. (2016, March 25). *Memory, Encoding Storage and Retrieval*. Retrieved from Simply Psychology: http://www.simplypsychology.org/memory.html

scream3r. (2016, April 5). *Maven Repository: org.scream3r >>jssc>> 2.8.0*. Retrieved from MVNRepository: http://mvnrepository.com/artifact/org.scream3r/jssc/2.8.0

SecondLife. (2016, March 24). *Second Life Official Site - Virtual Worlds, Avatars, Free 3D Chat*. Retrieved from Second Life: http://secondlife.com/

Seeed. (2015, December 28). *Find grovefamily - Buy grovefamily from Best grovefamily Distributors*. Retrieved from Seeed Studio: http://www.seeedstudio.com/depot/s/grovefamily.html

Sharma, A. K. (2016, March 28). *How does the magnet select button for Android Cardboard work?* Retrieved from Quora: https://www.quora.com/How-does-the-magnet-select-button-for-Android-Cardboard-work

Simulink, M. &. (2015, December 1st). *Foundations of Fuzzy Logic*. Retrieved from MathWorks: http://www.mathworks.com/help/fuzzy/foundations-of-fuzzy-logic.html

Simulink, M. &. (2015, December 5). *Fuzzy Inference Process*. Retrieved from MathWorks: http://www.mathworks.com/help/fuzzy/fuzzy-inference-process.html

Simulink, M. (2015, December 5). *What is Sugeno-Type Fuzzy Inference?* Retrieved from MathWorks: http://www.mathworks.com/help/fuzzy/what-is-sugeno-type-fuzzy-inference.html

Spradlin, L. (2016, March 28). *Google Cardboard Gets Play Store Collection, New SDKs, And Viewer Specifications*. Retrieved from Android Police: http://www.androidpolice.com/2014/12/10/google-cardboard-gets-play-store-collection-new-sdks-viewer-specifications/

Stanford. (2016, March 22). *Stemming and lemmatization*. Retrieved from NLP Stanford: http://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html

Strickland, J. (2016, March 24). *How Virtual Reality Works*. Retrieved from HowStuffWorks: http://electronics.howstuffworks.com/gadgets/other-gadgets/virtual-reality.htm

Strickland, J. (2016, March 24). *The Virtual Reality Environment - How Virtual Reality Works*. Retrieved from HowStuffWorks - Tech: http://electronics.howstuffworks.com/gadgets/other-gadgets/virtual-reality2.htm

Strickland, J. (2016, March 24). *The Virtual Reality Headset - How Virtual Reality Works*. Retrieved from HowStuffWorks - Tech: http://electronics.howstuffworks.com/gadgets/other-gadgets/virtual-reality4.htm

Strickland, J. (2016, March 24). *Virtual Reality Interactivity - How Virtual Reality Works*. Retrieved from HowStuffWorks - Tech: http://electronics.howstuffworks.com/gadgets/other-gadgets/virtual-reality3.htm

SWI-Prolog. (2015, December 10). *SWI-Prolog downloads*. Retrieved from swi-prolog: http://www.swi-prolog.org/Download.html

The National Autistic Society. (2016, February 20). *Sesory differences*. Retrieved from autism.org: http://www.autism.org.uk/sensory

TheInnovationAcademy. (2015, December 15). *Information Visualisation*. Retrieved from YouTube: https://www.youtube.com/watch?v=XqnslAnfh4Y

Unity. (2016, January 7). *Unity - Scripting API: Camera.fieldOfView*. Retrieved from docs.unity3d: http://docs.unity3d.com/ScriptReference/Camera-fieldOfView.html

Unity3D. (2016, March 26). *Unity - Manual: Coroutines*. Retrieved from docs.unity3d: http://docs.unity3d.com/Manual/Coroutines.html

Villarejo, M. V., Zapirain, B. G., & Zorrilla, A. M. (2016, February 23). *A Stress Sensor Based on Galvanic Skin Response (GSR) Controlled by ZigBee*. Retrieved from US National Library of Medicine National Institutes of Health: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3386730/

Wheatley, M. (2016, March 24). *DARPA using Oculus Rift to open a new front in cyberwarfare*. Retrieved from siliconANGLE: http://siliconangle.com/blog/2014/05/27/darpa-using-oculus-rift-to-open-a-new-front-in-cyberwarfare/

Wheatley, M. (2016, March 24). *Virtual Reality brings Big Data visualization to life.* Retrieved from siliconANGLE: http://siliconangle.com/blog/2014/05/29/virtual-reality-brings-big-data-visualization-to-life/

Wolf, H. (2016, March 27). *Virtual-Presence.org*. Retrieved from Virtual-Presence: http://virtual-presence.org/

Zhang, S. (2016, March 20). *The Neuroscience of Why Virtual Reality Still Sucks*. Retrieved from Gizmodo: http://gizmodo.com/the-neuroscience-of-why-vr-still-sucks-1691909123

A. Shleeg, Aishalaa M. Ellabib, I. (2013). Comparison of Mamdani and Sugeno Fuzzy Interference Systems for the Breast Cancer Risk. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, *7*(No. 10), 695 – 699.

Abdul Hamid, L., & Athiar Ramli, D. (2014). Quality Based Speaker Verification Systems Using Fuzzy Inference Fusion Scheme. *Journal of Computer Science*, *10*(3), 530–543. doi:10.3844/jcssp.2014.530.543

Baños, R. M., Botella, C., Alcañiz, M., Liaño, V., Guerrero, B., & Rey, B. (2004). Immersion and emotion: their impact on the sense of presence. *Cyberpsychology & Behavior : The Impact of the Internet, Multimedia and Virtual Reality on Behavior and Society*, *7*(6), 734–741. doi:10.1089/cpb.2004.7.734

Belkin, N., & Croft, W. (1992). Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, *29*(10), 1–10. doi:10.1145/138859.138861

Bosse, T., Duell, R., Hoogendoorn, M., Klein, M., van Lambalgen, R., van der Mee, A., … de Vos, M. (2009). A Generic Personal Assistant Agent Model for Support in Demanding Tasks. *Foundations of Augmented Cognition, Proceedings*, *5638*, 3–12. Retrieved from <Go to ISI>://WOS:000268101300001

Cádiz, R. F. (2006). Fuzzy-Logic Mapper Media Audiovisual. *Computer Music Journal*, *30*(No. 1), 67–82. Retrieved from http://www.jstor.org/stable/3682027?seq=4&loginSuccess=true#page_scan_tab_contents

Cingolani, P., & Alcalá-Fdez, J. (2012). JFuzzyLogic: A robust and flexible Fuzzy-Logic inference system language implementation. *IEEE International Conference on Fuzzy Systems*, 10–15. doi:10.1109/FUZZ-IEEE.2012.6251215

Cingolani, P., & Alcalá-Fdez, J. (2013). jFuzzyLogic: a java library to design fuzzy logic controllers according to the standard for fuzzy control programming. *International Journal of Computational Intelligence Systems*, *6*(sup1), 61–75. doi:10.1080/18756891.2013.818190

Dawson. C. (2008). Pratical research methods: A user-friendly guide to mastering research. *Vasa*, 153. Retrieved from http://medcontent.metapress.com/index/A65RM03P4874243N.pdf

Donalek, C., Djorgovski, S. G., Cioc, A., Wang, A., Zhang, J., Lawler, E., … Longo, G. (2014). Immersive and Collaborative Data Visualization Using Virtual Reality Platforms. *2014 IEEE International Conference on Big Data Immersive*, 609–614. doi:10.1109/BigData.2014.7004282

Edwards, J. (2011). Telepresence: Virtual Reality in the Real World. *IEEE Signal Processing Magazine*, (November), 9–13. doi:10.1109/MSP.2011.941853

Ehlert, P. (2003). Intelligent User Interfaces: Introduction and survey(Draft ver.). *Technology*, (February).

F. Wohlwill, J. (1970). EDRA02-Wohwill-340-344_1.pdf.

Hanani, U., Shapira, B., & Shoval, P. (2001). *Information Filtering: Overview of Issues, Research and Systems. CEUR Workshop Proceedings* (Vol. 1225). doi:10.1023/A

Heeter, C. (1992). Being there: The subjective experience of presence. *Presence: Teleoperators and Virtual Environments*, *1*(2), 262 – 271. doi:10.1109/VRAIS.1995.512482

Höök, K. (2000). Steps to take before intelligent user interfaces become real. *Interacting with Computers*, *12*(4), 409–426. doi:10.1016/S0953-5438(99)00006-5

Jassbi, J. J., Serra, P. J. A., Ribeiro, R. A., & Donati, A. (n.d.). Comparison of Mandani and Sugeno Inference Systems for a Space Fault Detection Application, 1–7. doi:10.1109/WAC.2006.376033

Kalawsky, R. S. (2000). The Validity of Presence as a Reliable Human Performance Metric in Immersive Environments. *Presence*, 1–16.

Kamboj, V., & Kaur, A. (2013). Comparison of Constant {SUGENO-Type} and {MAMDANI-Type} Fuzzy Inference System for Load Sensor. *International Journal of Soft Computing and Engineering (IJSCE)*, *3*(2), 204 – 207.

Kaur, A., & Kaur, A. (2012). Comparison of Mamdani-Type and Sugeno-Type Fuzzy Inference Systems for Air Conditioning System. *International Journal of Soft Computing and Engineering (IJSCE)*, *2*(2), 323 – 326.

Khayut, B., Fabri, L., & Abukhana, M. (2014). Intelligent User Interface In Fuzzy Environment. *International Journal of Artificial Intelligence & Applications (IJAIA), Vol. 5*(No. 1), 63–78.

Kölzer, A. (1998). Universal Dialogue Speci cation for Conversational Systems.

Negnevitsky, M. (2005). *Artificial Intelligence A Guide to Intelligent Systems* (Second Edi.). Tasmania, Australia.

Norman, D. A. (2002). *The Design of Everyday Things*. *Human Factors and Ergonomics in Manufacturing* (Vol. 16). doi:10.1002/hfm.20127

Paraiso, E. C., & Barthès, J.-P. A. (2006). An intelligent speech interface for personal assistants in R&D projects. *Expert Systems with Applications*, *31*(4), 673–683. doi:10.1016/j.eswa.2006.01.018

Rheingold, H. (2011). Virtual Reality. *Cryptologic Quarterly*, 21–50. doi:392132

Rigby, S., & Richard, R. (2011). Immersion and Presence. *Glued to Games: How Video Games Draw Us In and Hold Us Spellbound*, 81–96. Retrieved from http://www.ism.univmed.fr/mestre/projects/virtual reality/Pres_2005.pdf

Salman AbdulWahed, M., & Seno Ismat, N. (2012). A Comparison of Mamdani and Sugeno Inference Systems for a Satellite Image Classification. *Anbar Journal for Engineering Sciences* □□□□ □□□□□□□□ □□□□□□□ □□□□□□□□□, □□□□□□□□ □□(الجزء الثاني - الخاص العدد), 296–306.

Sergieh, H. M., Gianini, G., D{"o}ller, M., Kosch, H., Egyed-Zsigmond, E., & Pinon, J.-M. (2012). Geo-based Automatic Image Annotation. *Icmr*, 46. Retrieved from http://dl.acm.org/citation.cfm?id=2324850

Slater, M., & Wilbur, S. (1997). A Framework for Immersive Virtual Environments (FIVE): Speculations on the Role of Presence in Virtual Environments. *Presence: Teleoperators and Virtual Environments*, *6*(6), 603–616. doi:10.1007/s10750-008-9541-7

Steuer, J. (1992). Defining Virtual Reality: Dimensions Determining Telepresence. *Journal of Communication*, *42*(4), 73–93. doi:10.1111/j.1460-2466.1992.tb00812.x

Sumalatha, V. M., Ramani, K. V., & Lakshmi, K. V. (2011). Fuzzy Inference System to Control PC Power Failures. *International Journal of Computer Applications (0975-8887)*, *28*(4), 10–17.

Tufte, E. R. (2001). *The Visual Display of Quantitative Information*.

Valenzuela-Rendón, M. (n.d.). The Fuzzy Classifier System: A Classifier System for Continuously Varying Variables. Monterrey, N.L.: Centro de Inteligencia Artificial.

Vande Moere, A. (2008). Beyond the tyranny of the pixel: Exploring the physicality of information visualization. *Proceedings of the International Conference on Information Visualisation*, 469–474. doi:10.1109/IV.2008.84

Zaher, H., Kandil, A. E., & Fahmy, R. (2014). SCIENCEDOMAIN international Comparison of Mamdani and Sugeno Fuzzy Inference Systems for Prediction (With Application to Prices of Fund in Egypt), *4*(21), 3014–3022.