POLITECNICO DI TORINO

Master's Degree in Cinema and Media Engineering

Politecnico di Torino

Master's Degree Thesis

Development of a horror game with Emotibit sensor integration for EDA measurements

Supervisors

Prof. Andrea BOTTINO

Prof. Francesco STRADA

Prof. Klaas BOMBEKE

Dr. Aleksandra ZHELEVA

Candidate

Edoardo PELOSIN

April 2023

A.A. 2023/2024

Abstract

This thesis endeavors to conduct a comprehensive and detailed examination, along with documentation, of the development of a virtual reality (VR) application. Additionally, it aims to explore the process involved in integrating an offline feedback loop designed for the analysis of physiological data. The primary goal of this research endeavor is to undertake the preprocessing of raw physiological data, focusing particularly on the Electrodermal Activity (EDA), which is the variation of the electrical conductance of the skin in response to sweat secretion. The EDA signal can be reflective of the intensity of our emotional state, otherwise known as emotional arousal. Our level of emotional arousal changes in response to the environment and setting we're in, if something is scary, threatening, joyful, or otherwise emotionally relevant, then the subsequent change in emotional response that we experience also increases eccrine sweat gland activity. EDA is gathered (with a werable physiological sensor called Emotibit) from participants actively engaged in the VR experience. Notably, the VR application in question is designed to induce stress in its users. This deliberate induction of stress serves as a catalyst for the observation and analysis of variations in EDA levels, thereby illuminating potential correlations between emotional arousal and physiological responses within the VR environment. The inception of this scholarly pursuit can be traced back to the formative stages of a VR application developed during the course of an internship. This internship was undertaken at MICT imec-mict-UGent, a Research Group for Media, Innovation, and Communication Sciences at Ghent University.

Table of Contents

List of Tables

List of Figures

Acronyms

EDA

Electrodermal Activity

VR

Virtual Reality

MICT

imec-mict-UGent, a Research Group for Media, Innovation, and Communication

HMD

Head-Mounted Display

FoV

Field of View

QoE

Quality of Experience

SAM

Self-Assessment Manikin

PPG

Photoplethysmography

IMU

Inertial Measurement Unit

SCR

Skin Conductance Response

SCL

Skin Conductance Level

GSR

Galvanic Skin Response

OSC

Open Sound Control

MSE

Mean Square Error

CC

Correlation Coefficient

Chapter 1

Introduction

Virtual reality (VR) has become an intriguing avenue for exploring user experiences and enhancing the quality of interactions within digital environments. This master's thesis embarks on an exploration of the challenges surrounding the integration of an offline feedback loop within a Unity application, focusing specifically on the context of an immersive VR horror experience. Additionally, it aims to establish a correlation between electrodermal activity (EDA) and user fear levels. As the VR application in question elicits frightening events and measures (via the Emotibit sensor) user EDA, self-assessment questionnaires from participants will aid in the analysis of EDA data. The journey of this investigation was initiated during an internship at MICT. The Unity application conceived for this thesis originates as a derivative of the one developed during the aforementioned internship. It forms the foundation for the experimental segment of Aleksandra Zheleva's PhD research, which pertains to the quality of user experience in VR.

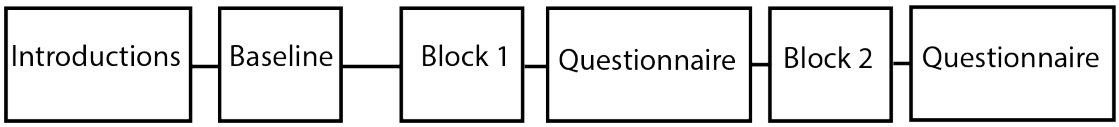

Figure 1.1: Structure of the experiment

The VR horror experience is designed around a kitchen setting, where users are presented with instructions through a television display. The task is to tidy up the virtual kitchen by discarding broken items and organizing the remaining ones on the shelves. The experiment unfolds through two blocks, each spanning 12 minutes. One block introduces frightening events, while the other remains non-threatening, both still requiring participants to tidy the virtual kitchen. Post each block, participants are prompted to complete a questionnaire addressing their immersive experience, frightening, and arousal. Frightening events are integrated into one of

the blocks, manifesting as auditory (A#), visual (V#), or audio-visual (A#V#) stimuli. The sequence of these events is counterbalanced to eliminate order-induced biases in subjective (questionnaire) and objective (EDA) data collection. These frightening events include scenarios such as encountering a zombie emitting screams (V1) or an eerie figure with corresponding auditory cues (A2), resulting in 12 potential combinations distributed within a two-minute time range.

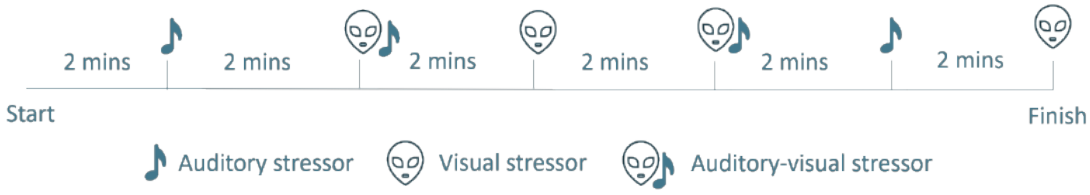

Figure 1.2: Block with frightening events

1.1 Quality of Experience (QoE) and Virtual Reality PhD

The PhD project "Could You Repeat That, Please? Increasing The Quality And Replicability Of Physiological Virtual Reality Research Through A Comprehensive Experimental Framework" aims to construct a comprehensive research framework. This framework encompasses a model of user experience factors in VR, a survey of applicable physiological sensors and their measured factors, and a data collection and processing mechanism. To validate this framework, a series of experiments with end users is planned. The Unity application developed in this thesis is integral to achieving this aim. The QoE model resulting from this PhD project encapsulates 252 factors categorized into Context, System, Content, and User branches. These factors span aspects such as physical environment, technical components, storytelling elements, and user characteristics. This model aims to enhance VR experiences by offering insights into factors affecting user engagement and immersion.[1]

Context factors

These factors pertain to the situational properties of the physical environment in which the user is engaged in VR. This includes aspects such as size and layout of the playing area, location, and temperature. Safety is an important sub-branch within context factors, encompassing physical, psychological, ethical, and data dimensions. Additionally, factors like purpose for use, cost (process, relationships, and economic), and accessibility are considered.

System factors

This category of factors is primarily associated with the technical components of VR. It includes hardware components like head-mounted display (HMD) resolution, haptic devices, field of view (FoV), and tracking devices. Software developments such as image rendering algorithms and tracking algorithms are also considered. Network characteristics like latency, bandwidth, bit rate, and delay play a role in determining the VR experience.

Content factors

This branch focuses on the elements related to storytelling in VR. It addresses how to guide the user's intention in VR without overwhelming them with cues. The design of the soundscape to enhance user immersion in the story is another aspect. Interaction design in the virtual environment is also a key consideration within content factors.

User factors

This category encompasses factors that relate to the user's characteristics and experiences. It includes static factors such as age, previous experience, physical health, and other personal traits. Dynamic factors like presence (feeling of "being there" in the virtual environment), cybersickness (motion sickness or discomfort in VR), immersion, and enjoyment are also important user-related considerations.

1.2 Role of the Unity application in the PhD

In the context of the PhD project outlined in 1.1 "Quality of Experience (QoE) and Virtual Reality", the Unity application detailed in the preceding section serves a pivotal role in the advancement of research. Its primary objective is to rigorously assess and validate the QoE (Quality of Experience) model previously elucidated, through a series of experiments involving end users. The crux of this application's purpose lies in the investigation of the intricate interplay between distinct user-centric factors (namely, immersion, presence, arousal, and valence) and content-related factors such as multimodal storytelling, encompassing auditory, visual, and audio-visual cues. These elements are scrutinized within the context of two experiments, aimed at delving deep into the realm of VR (Virtual Reality) immersiveness. This exploration entails an exhaustive analysis, considering both subjective and objective dimensions of data. Subjective data collection hinges on the application of the Self-Assessment Manikin (SAM) questionnaire and an immersion questionnaire, both tailored to gauge participants' perceptions and experiences. In parallel, objective data collection involves the meticulous monitoring of participants'

electrodermal activity (EDA) utilizing Emotibit hardware. The ExperienceDNA framework is strategically deployed to simultaneously record and synchronize both subjective and objective data throughout participants' engagement with the VR environment. The initial experiment pertains to the introduction of diverse stimulus types (auditory, visual, and audio-visual) within the VR setting. Its core objective is to untangle the intricate dynamics through which different sensory inputs influence not only immersiveness but also arousal and valence within the VR context. The study draws on a dual perspective, meticulously analyzing the impact of these stimuli through both subjective user feedback and quantifiable objective measures, thus contributing to a profound comprehension of the constituents underlying captivating and pleasurable VR experiences. Guided by the insights gleaned from the initial experiment, the trajectory advances toward the conception of an adaptive VR environment in the subsequent experiment. This environment, distinguished by its employment of real-time feedback loops, harnesses dynamic real-time data to enrich immersiveness and user engagement. The fusion of this concept with the established ExperienceDNA framework culminates in a personalized VR encounter. The core premise of this second experiment revolves around the exploration of how the incorporation of feedback loops can engender responsive and tailored modifications within the VR environment. This investigation aspires to discern how these malleable adjustments resonate in terms of overall immersiveness and emotional resonance within the VR experience. The potential ramifications of these findings extend to the conceptualization and creation of VR environments that adapt organically to user interactions.

1.3 ExperienceDNA

The Unity application described in this thesis utilizes the ExperienceDNA framework to create an immersive experience that logs both subjective and objective data streams. ExperienceDNA is a research framework that can be used to create immersive VR experiences for user testing, control the interactions between users and their surroundings, and collect a wide range of fine-grained sensor data, allowing researchers to map the entire user experience. The framework is developed in Unity and supports both qualitative and quantitative data logging, capturing various data such as eye gaze, pupil dilation, blink rate, object interaction, reaction times, accuracy, and heart rate. The data can be monitored in real-time and exported at the end of the VR experience [2]. For this thesis, a variation of the application utilized in Alecksandra Zheleva's PhD project has been developed, integrating the pre-processing algorithm for EDA data on the Vive Focus 3 headset. This alteration introduces offline feedback loops as the primary point of differentiation between the two applications.[3]

1.4 Objectives

This thesis undertakes the creation of a VR application within Unity, enriched by an offline feedback loop for analyzing raw EDA data. The application is designed for the standalone device, Vive Focus 3. The primary objective of this thesis is to analyze Electrodermal Activity (EDA) measurements collected during Virtual Reality (VR) experiences, compare them with participants' self-assessment questionnaires, and determine if EDA data confirm that participants who report being more frightened by the VR experience exhibit higher EDA peaks. The secondary objective is to simplify and adapt a Python algorithm from the PhD project, enabling it to be executed on the Vive Focus 3, thus integrating the feedback loop. This innovation seeks to eliminate the necessity of a remote server and enhance the accessibility and applicability of the feedback loop within the VR application.

1.5 Organization of chapters

The thesis is structured into six chapters. The first chapter provides an elucidation of the context in which the Unity application was developed. It also highlights the application's significance within the broader scope of the PhD research, titled "Quality of Experience and Virtual Reality: An Exhaustive Literature Review." Furthermore, this chapter outlines the general structure and framework of the Unity application. Moving on to the second chapter, it involves an analysis and comparison of various use cases pertaining to feedback loops integrated into virtual reality applications and methodologies for analyzing and interpreting EDA. In the third chapter, comprehensive details are presented regarding the software, hardware, and protocols employed to develop the Unity application. This section provides a comprehensive overview of the technological elements that have contributed to the application's development. The fourth chapter is about the conducted tests, specifically designed to evaluate and measure the efficiency and precision of the feedback loop integrated within the Unity application. The fifth chapter examines EDA measurements conducted in 16 experiments and seeks correlations between the level of fear expressed by the user through post-experience self-assessment questionnaires and EDA data. Lastly, the sixth chapter presents an exposition of the results obtained from the conducted measurements. Additionally, a comprehensive analysis and interpretation of these results are provided. Furthermore, there will be a contemplation on potential future advancements of the application, where the offline feedback loop and the EDA dara are utilized to adapt the VR environment based on the user's emotional state.

Chapter 2

State of art

This chapter encompasses an analysis of research papers instrumental in attaining the objectives of the thesis project.

2.1 Affective Feedback in a Virtual Reality based Intelligent Supermarket

The paper discusses a VR experiment with a design and structure resembling that of the current project thesis experiment. The experiment design involves the utilization of a virtual reality based experimental platform to investigate the integration of continuous EDA measurements into a context-aware intelligent environment (CAIE). Participants navigate a virtual supermarket in a fully position-tracked space while wearing an Empatica E4 wristband, which continuously acquires real-time physiological data. The user is helped by a navigation-assist service that aims to find the shortest direct path for customers based on a dynamically changing shopping list. Participants, equipped with VR glasses and positioned in a fully tracked space, receive a pre-populated shopping list of 10 items. The system displays item numbers as overlays in the VR scene, creating the illusion of a dynamically changing list. The navigation-assist service guides participants through green path arrows, with a red arrow indicating the final destination. The experiment is conducted in a Wizard-of-Oz fashion, where the experimenter activates path arrows based on participants' spoken item numbers. Paths include both direct routes and deliberately winding ones to induce techno-stress and assess user reactions. Participants are instructed to follow the indicated path, even if not a direct route [4]. The experiment includes correct services (CS: direct path) and wrong services (WS: winding path) for different items on the list. Additionally, a Paced Stroop Test (PST) experiment is conducted with a task pacing time of 3 seconds between Stroop figures, totaling 180 seconds.[4][5]

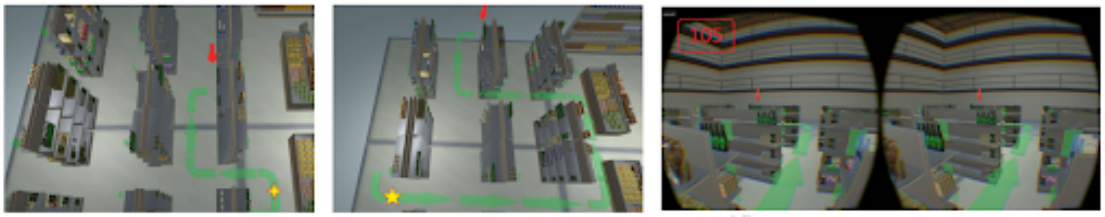

Figure 2.1: VR scene

2.2 Virtual Reality Experiments with Physiological Measures

This paper establishes an academic standard protocol for VR experiments involving physiological measures. The protocol involves a systematic series of steps for the recruitment, preparation, and execution of participants in the context of conducting virtual reality (VR) experiments with physiological sensors, adhering to the Experiments in Virtual Environments (EVE) framework. The following is a comprehensive breakdown of the protocol:

Recruit and Prepare Participants

- Select participants based on specific demographics through established recruitment systems or mailing lists.
- Contact participants via email, providing session details and necessary instructions regarding attire, dietary restrictions, and lifestyle activities preceding the experiment.

Prepare the Experiment and Physiological Devices Using EVE

- Initiate the requisite technical components before each experimental session.
- Ensure functionality of electrodermal activity (EDA) and electrocardiography (ECG) measurement devices.
- Utilize EDA/ECG software (compatible with EVE) to configure settings and synchronize with physiological sensors.
- Prepare the experimental environment, including room conditions and control interfaces.

Experimental Procedure

- Commence the session with participant introduction, consent procedure, and briefing on the experimental timeline.
- Establish connections for EDA and ECG sensors, adhering to specific protocols for sensor placement and skin preparation.
- Administer pre-experiment questionnaires, ensuring participant comfort and understanding.

Joystick Training and Baseline Video

- Provide participants with training on joystick usage through instructional videos and practical exercises.
- Instruct participants to complete training tasks, including maze navigation and gem collection.
- If applicable, introduce sound elements with headphones and initiate viewing of a baseline nature video.

Navigation Task

- Confirm participants' understanding of navigation task instructions.
- Prompt participants to commence the navigation task using the joystick trigger.

Final Physiological Measures and Detachment of Physiological Sensors

- Conclude the experiment with final blood pressure measurement and recording cessation for EDA and ECG.
- Safely detach the blood pressure cuff and EDA electrodes.
- Remove the joystick and headphones.

Post-Experiment Questionnaires

- Administer post-experiment questionnaires using a computer interface, covering various aspects such as frightening level, self-assessment, and simulator sickness.

End of the Experimental Session

- Conclude the experimental phase, express gratitude to participants, and address any queries.
- Facilitate participant payment and obtain necessary signatures on receipts.

After Each Experimental Session

- Conduct experiment diagnostics through the EVE framework, replaying trajectories for analysis.
- Mark events in physiological measurement files for precise data analysis.
- Save physiological measurement files and export experimental data for backup.
- Power down EDA/ECG machines and clean electrodes for subsequent use.
- Update participant attendance in the recruitment system.

This detailed protocol aims to ensure consistency, reliability, and efficiency in the execution of VR experiments with physiological sensors [6].

2.3 Adaptive virtual reality

The paper outlines the three main processes of the closed-loop approach to adaptive Virtual Reality (VR). This approach is also employed in the current thesis project to visualize processed EDA data. The three procedures of the closed-loop approach:

- Signal acquisition. This phase is distinguished by the real-time gathering of unprocessed data, including subsequent data "cleaning" operations such as artifact rejection or correction.
- Data analysis and inference. Within a specific time frame or epoch, the refined data is subjected to analysis, culminating in the derivation of inferences about the user's condition, such as the categorization of their emotional state.
- A repertoire of adaptive interactive mechanisms (AIM). These mechanisms are designed to promote specific behavior and/or experiences. Further details on these processes can be found in the references [7][8][9].

The configuration and nature of these adaptations are directed by a defined standard or predefined objective, such as the enhancement of task engagement, the promotion of relaxation, or the modulation of frightening levels. Consequently, these adaptations impact the state and behavior of the individual user, thereby serving

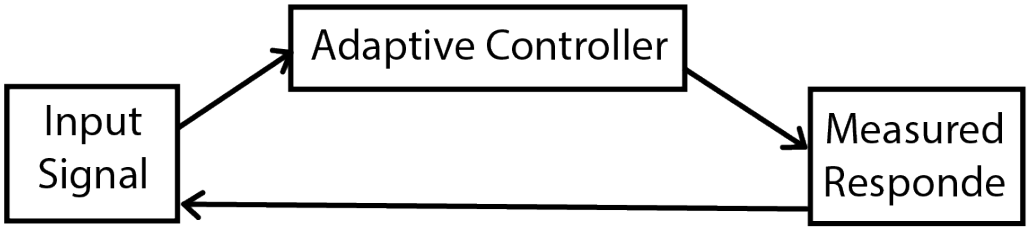

Figure 2.2: Closed loop system

as a subsequent input signal that, in turn, completes the closed-loop framework. The realization of Adaptive VR hinges on this integration of real-time monitoring and adaptation within a closed-loop system. establish a formal linguistic framework and systematic categorization pertaining to the constituent components of the feedback loop. It is intended to contribute to the broader comprehension of virtual reality (VR) experiments involving physiological sensors, facilitating a structured examination of the project [10].

2.4 D-flow: immersive virtual reality and real-time feedback for rehabilitation

This paper delineates the logical approach of the real-time feedback loop implemented by the D-Flow software. D-Flow is a sophisticated software system crafted for the purpose of fostering the advancement of interactive and immersive virtual reality applications, with a particular focus on applications within the realms of clinical research and rehabilitation. Its core functionality resides in facilitating the establishment of real-time feedback loops, wherein the behavioral patterns of subjects are meticulously gauged through the utilization of multi-sensory input devices. Subsequently, the system orchestrates the delivery of comprehensive motor-sensory, visual, and auditory feedback to the subjects, thereby engendering an enriched and responsive virtual environment. This software system is underpinned by a versatile and expansible application development framework, which empowers operators with the capability to delineate intricate feedback strategies through the medium of visual programming.[11] The D-Flow kernel is the framework in which the actual virtual reality applications can be defined. The operational paradigm of the D-Flow kernel is based on the main processing loop, characterized by a frame-by-frame modus operandi. Each discrete frame encompasses a meticulously choreographed sequence of four principal steps, delineated as follows:

Await Signal from Real-Time Device Manager (Optional)

The initial step involves the kernel awaiting a signal from the real-time device manager, signaling the availability of new data from a specific device. It is important to note that this step is optional and contingent upon the real-time requirements of the application.

Module-Specific Actions and Event Broadcasting

Subsequently, for each module, the kernel engages in the execution of module-specific actions tethered to events previously scheduled for broadcast during the antecedent frame. This phase constitutes a very important aspect of the operational cycle, ensuring the synchronization of module activities with the temporal framework of the virtual environment.

Internal Processing of Modules

The third step encompasses the internal processing of each module, encompassing the following sub-steps

- Updating user interface parameters as deemed appropriate for the evolving context.
- Generating new output data predicated upon the available input data.
- Effecting modifications to objects within the virtual reality environment to reflect the dynamically evolving state.
- Strategically scheduling the broadcast of a plethora of global events anticipated to transpire in the forthcoming frame.

Virtual Reality Environment Modification and Display Update

All operations influencing alterations in the virtual reality environment are meticulously managed through the Distributed Rendering System. This component assumes responsibility for handling the intricacies of virtual environment modification, ensuring seamless updates to displays in consonance with the processed data. The sequencing of module processing adheres to a structural hierarchy, wherein the order of execution is contingent upon the interconnection arrangement. In instances of loops, modules reliant on streamed input data from the synchronized data buffer are accorded priority during processing, ensuring the systematic and synchronized evolution of the computational tasks within the framework of the D-Flow kernel. [11]

Chapter 3

Development of the Unity application and tools used

3.1 Emotibit

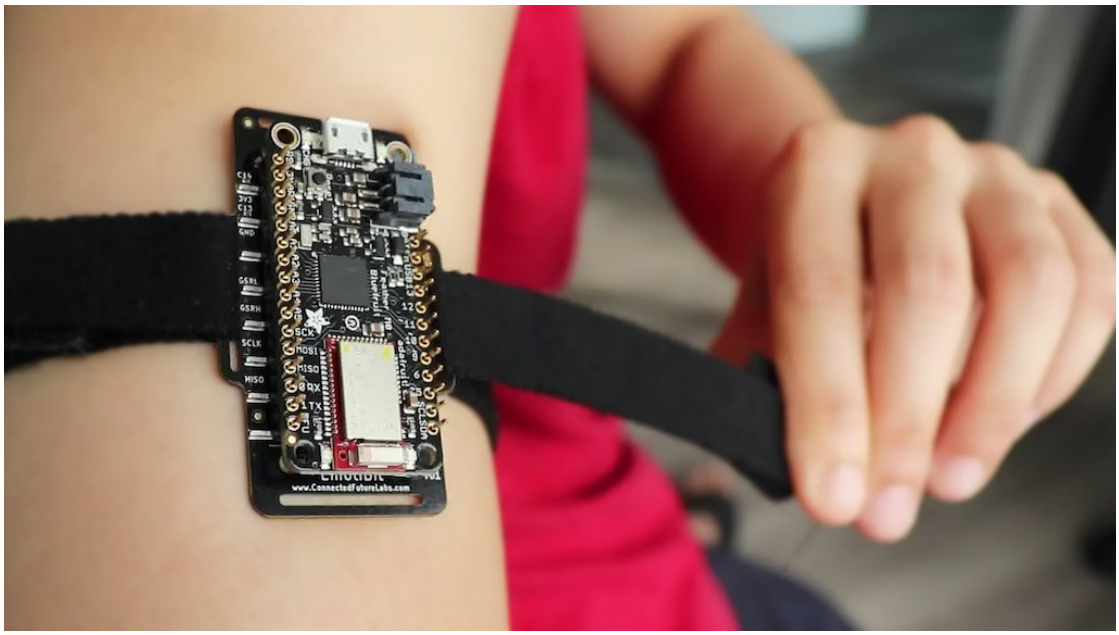

Figure 3.1: Emotibit

EmotiBit possesses the capability to wirelessly transmit and locally store data originating from a diverse array of sensors, constituting a multi-modal constellation.

This constellation comprises Electrodermal Activity (EDA), multi-wavelength Photoplethysmography (PPG), a temperature sensor of medical-grade quality, a 9-axis Inertial Measurement Unit (IMU), alongside an expanding repertoire of derivative metrics. Specifically, EmotiBit employs a documented circuit architecture for EDA measurement, meticulously calibrated during fabrication to furnish precise readings across a broad spectrum of skin conductance levels, thereby ensuring compatibility with various skin types. [12] Given that Emotibit's EDA sensor operates by detecting variations in electrical (ionic) activity attributable to fluctuations in sweat gland function, the electrodes necessitate sensitivity to such changes and efficacy in relaying this data to the recording apparatus. Emotibit's EDA electrodes feature an Ag/AgCl (silver-chloride) contact interface with the skin. The selection of Ag/AgCl electrodes is predicated upon their cost-effectiveness, durability, safety for human contact, and capacity to faithfully transmit signals stemming from ionic activity. EDA data is sampled at a rate of 15 Hz and quantified in micro-Siemens. Subsequently, the forthcoming subsection will delineate the nature of EDA and elucidate its acquisition and transmission via the Emotibit sensor.

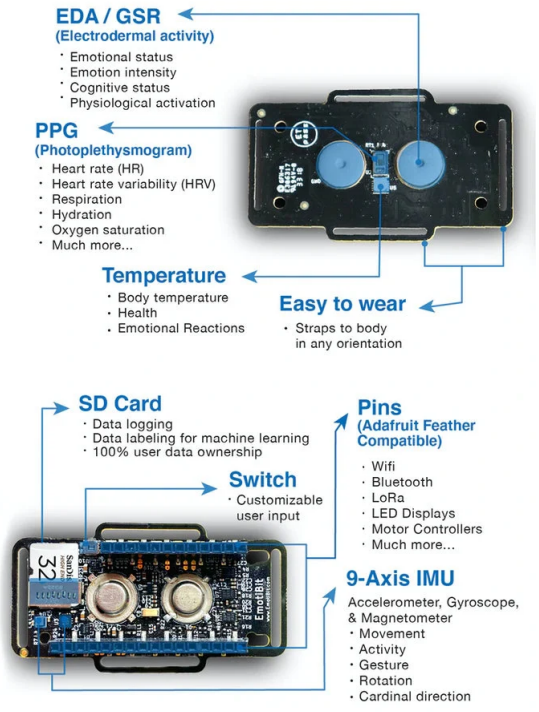

Figure 3.2: Emotibit's sensors

3.1.1 EDA signal

Electrodermal activity (EDA), also known as galvanic skin response (GSR), denotes the alteration in skin electrical conductance triggered by sweat secretion, often in minute quantities. This physiological data is gathered through the application of a low, imperceptible, and constant voltage to the skin, followed by the assessment of variations in skin conductance. EDA devices, employing electrodes affixed to the skin, facilitate this measurement of the electrical signal. While EDA is implicated in the regulation of internal temperatures, its robust correlation with emotional arousal has been extensively documented in research. Signals originating from the sympathetic nervous system precipitate changes in Skin Conductance Response (SCR), a primary focus of researchers. The EDA signal serves as an indicator of emotional arousal intensity, which dynamically shifts in response to environmental stimuli – be they alarming, joyful, or emotionally salient. Investigations have elucidated how both positive and negative stimuli elicit heightened arousal and increased skin conductance, reflecting various dimensions of emotional response. Consequently, the EDA signal does not categorically delineate emotional types but rather signifies their intensity. The temporal profile of the signal is construed as the outcome of two concurrent processes: a tonic base level driver characterized by slow fluctuations (seconds to minutes), and a phasic component exhibiting rapid variations within seconds. Alterations in phasic activity manifest as discernible bursts within the continuous data stream, typified by steep inclines to distinctive peaks followed by gradual declines relative to the baseline level.

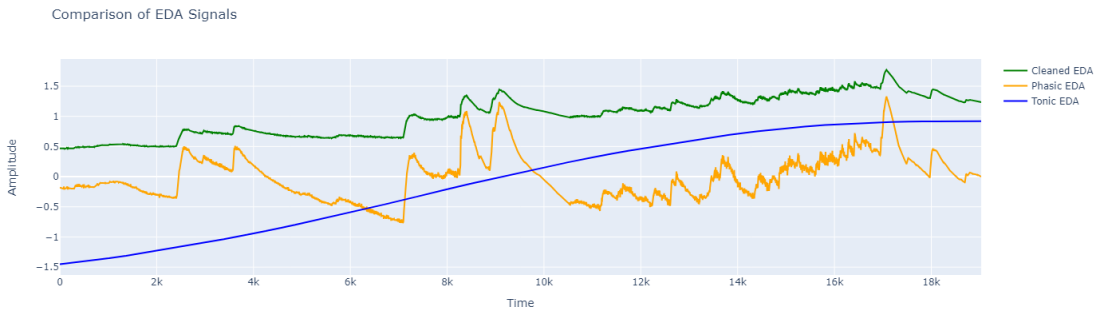

Figure 3.3: Graph of raw EDA signal, tonic EDA component, phasic EDA component

Researchers concentrate on the latency and amplitudes of phasic bursts relative to stimulus onset when investigating alterations in the EDA signal prompted by sensory stimuli such as images, videos, and sounds. When substantial modifications occur in EDA activity following a stimulus, it is designated as an Event-Related Skin Conductance Response (ER-SCR). These responses, commonly termed EDA peaks,

offer insights into emotional arousal triggered by stimuli. Additionally, peaks in EDA activity unrelated to stimulus presentation are labeled as Non-Stimulus-locked Skin Conductance Responses (NS-SCR). [5]

3.1.2 OSC protocol

The Emotibit records electrodermal activity (EDA) data at a frequency of 15 times per second, employing floating-point encoding for representation. A Wi-Fi connection links the Emotibit device to a personal computer (PC) hosting specialized software for managing the Emotibit data stream (called Emotibit Oscilloscope), facilitating data transmission from Emotibit to the PC.

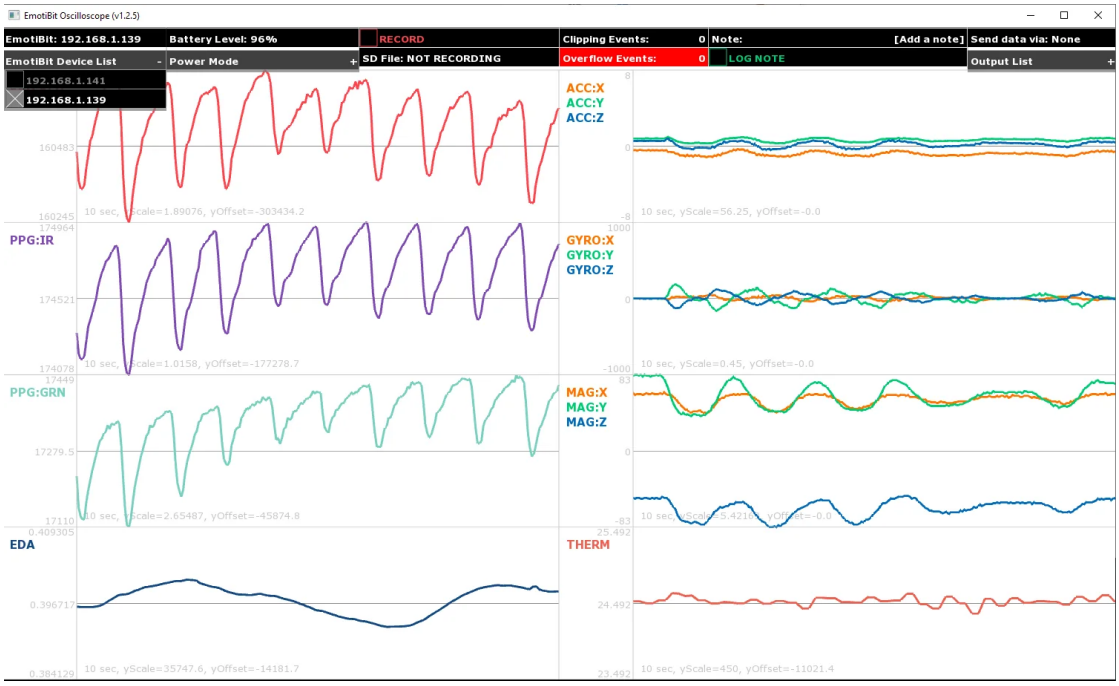

Figure 3.4: software interface for managing the Emotibit data stream

Subsequently, the PC streams this data through the Wi-Fi network to a predetermined Wi-Fi port designated in the oscOutputSettings XML configuration file. An OpenSoundControl (OSC) message, bearing the OSC address "/EmotiBit/0/EDA", is dispatched by the PC to encapsulate the EDA data. Concurrently, the virtual reality (VR) headset actively monitors the specified Wi-Fi port to receive and visualize the electrodermal activity (EDA) data within the Unity application environment.

```
<patchboard>
    <settings>
        <input>
            <type>EmotiBit</type>
        </input>
        <output>
            <type>OSC</type>
            <ipAddress>10.27.91.226</ipAddress>
            <port>12346</port>
        </output>
    </settings>
    <patchcords>
        <patch>
            <input>EA</input>
            <output>/EmotiBit/0/EDA</output>
        </patch>
    </patchcords>
</patchboard>
```

Figure 3.5: XML file of the OSC settings

Open Sound Control (OSC) is an open, transport-independent, message-based protocol developed for communication among devices. OSC employs an open-ended, URL-style symbolic naming scheme, facilitating the symbolic designation of parameters. This characteristic enhances the ease of identifying and managing various attributes of sound and multimedia devices. Furthermore, OSC offers symbolic and high-resolution numeric argument data encoding. A distinctive feature of OSC is its inclusion of a pattern matching language, enabling the specification of multiple message recipients within a single communication. This functionality enhances communication flexibility and versatility. Moreover, OSC enables the routing of messages to multiple procedures on an OSC server, facilitating intricate and distributed control over the devices. OSC supports high-resolution time tags utilizing 64-bit time values, enabling precise event timing and synchronization. The protocol also supports message bundling, allowing for the aggregation of multiple messages into bundles for simultaneous execution, ensuring synchronized actions

across devices. In summary, OSC presents a flexible and robust protocol for real-time control of multimedia devices, boasting a comprehensive suite of features for communication and data encoding.

3.2 HTC Vive Focus 3

The VR headset utilized for the execution of the application is the VIVE Focus 3, as requested by the MICT research team with whom this research project originated. This VR headset, developed by HTC and introduced to the market in 2021, stands as

Figure 3.6: VIVE Focus 3

a pivotal virtual reality (VR) headset engineered with a distinct focus on enterprise applications. Its design embodies several noteworthy technical specifications and design elements essential for immersive user experiences. Featuring dual 2.88-inch LCD panels, the Vive Focus 3 delivers a combined resolution of 2448 x 2448 pixels per eye, culminating in an impressive total resolution of 4896 x 2448 pixels and a refresh rate of 90Hz. Driving the device's performance is the Qualcomm Snapdragon XR2 Platform, crafted for the demands of extended reality (XR) applications. Boasting 8GB of RAM and 128GB of onboard storage, the headset provides substantial resources for seamless operation and content storage. Noteworthy is the implementation of inside-out tracking facilitated by four cameras, a feature integral to the Vive Focus 3's design. This tracking system eliminates the necessity for external sensors while maintaining precise positional tracking. Furthermore, the headset integrates controllers tailored for enhanced ergonomics and superior tracking precision. Complementing these features are removable open-back speakers, certified for high-resolution spatial audio.

3.3 ExperienceDNA framework

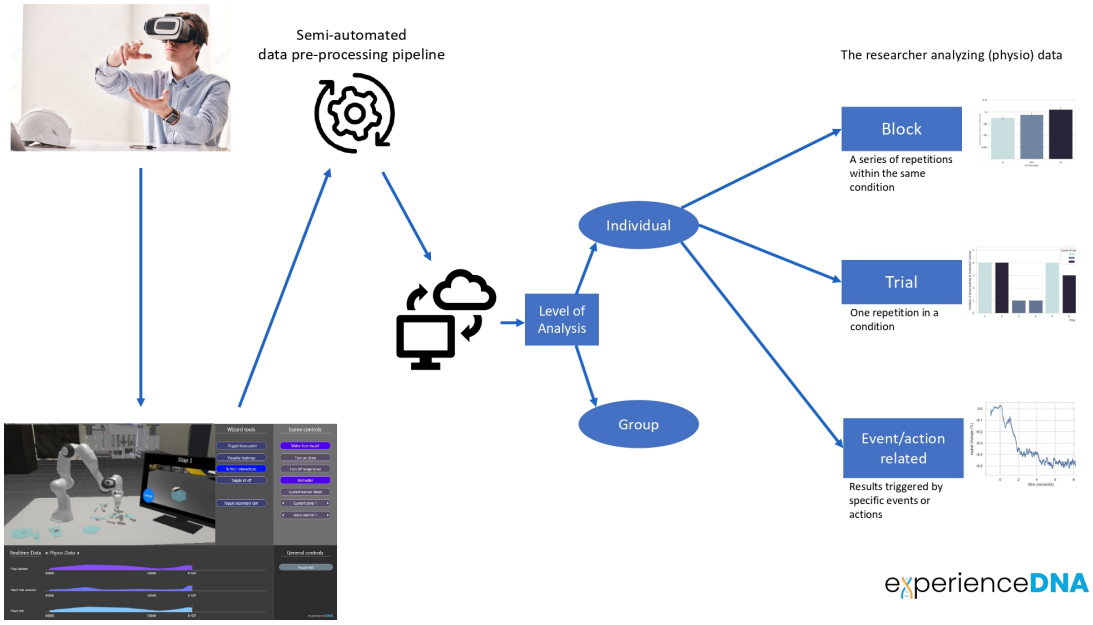

Figure 3.7: Architecture of the ExperienceDNA framework

In the development of the Unity application is used the the ExperienceDNA framework (developed in Unity by the MICT research team), which makes the structure of the virtual reality experiment more modular and easily mappable and modifiable.[2] The ExperienceDNA framework allows to create immersive experiences for product testing, control the interactions between users and their surroundings, and collect a wide range of fine-grained sensor data. All of which results in a user experience that can be mapped in detail. Most notably, the ExperienceDNA framework facilitates both qualitative and quantitative data logging. This means that the user experience can be fully mapped via capturing eye gaze data, pupil dilation data, blink rate data, object interaction data, reaction times, accuracy and heart rate monitoring. In the future, the framework will accommodate various other sensors, such as EEG, to facilitate a fine-grained user experience profile. The data can be both monitored in real-time and logged and exported at the end of the VR experience. This framework facilitates the deconstruction of experiments into blocks, with each block containing trials, and each trial featuring characteristic events. Consequently, researchers analyzing the physiological data, gathered during the experience, can conduct macro analyses on the block data or focus on individual events. The physiological data are captured by the sensor worn by the user and streamed to the VR headset. The headset

transmits the raw data to a server where Python and MATLAB algorithms are applied for pre-processing. Subsequently, the data are aggregated into groups.[2] In this project, the ExperienceDNA framework is employed solely for establishing the VR experience's structure and not for transmitting data to a server for processing. One of the primary objectives of this project thesis is to translate certain pre-processing Python algorithms mentioned above into C# algorithms, enabling real-time execution within the Unity application. The VR experience adheres to a block structure defined within a JSON file (referred to as "data_structure.json"), which is parsed using methods provided by the ExperienceDNA library.

```
{
  "Id": -1,
  "Name": "Tetra Telenet",
  "Description": "",
  "DurationBetweenBlocks": 0.0,
  "DurationBetweenTrials": 0.0,
  "Blocks": [
    {
      "LinkedTrials": [
        146,
        1,
        2,
        3,
        4,
        5,
        6,
        158
      ],
      "LinkedConditions": [
        6
      ],
      "IsBreak": false,
      "ConfigId": 1,
      "Name": "Block 1 (User1)",
      "Description": ""
    },
    {
      "LinkedTrials": [
        182,
        7,
        8,
        9,
        10,
        11,
        12,
        159
```


Figure 3.8: data_structure json file

Each block is characterized by specific conditions (user number) and comprises eight trials. Each trial presents a condition that may include a frightening stimulus (A#, V#, A#V#) or a non-stimulus (NS). In the initial scene, termed "MainMenu," participants select their assigned user number, previously communicated by the research team. Based on the entered user number, one of the stimulus sequences

corresponding to a pair of blocks (one without frightening stimuli and one with frightening stimuli) is selected. There are twelve pairs of blocks defined within the JSON file "data_structure" that identify 12 different sequences of stimuli.

	Block1						Block2					
	Trials											
	A1	V1	V2	A1V1	A2V2	A2	NS	NS	NS	NS	NS	NS
	V1	A1V1	A1	A2	V2	A2V2	NS	NS	NS	NS	NS	NS
	A1V1	A2	V1	A2V2	A1	V2	NS	NS	NS	NS	NS	NS
	A2	A2V2	A1V1	V2	V1	A1	NS	NS	NS	NS	NS	NS
	A2V2	V2	A2	A1	A1V1	V1	NS	NS	NS	NS	NS	NS
	V2	A1	A2V2	V1	A2	A1V1	NS	NS	NS	NS	NS	NS
	NS	NS	NS	NS	NS	NS	A1	V1	V2	A1V1	A2V2	A2
	NS	NS	NS	NS	NS	NS	V1	A1V1	A1	A2	V2	A2V2
	NS	NS	NS	NS	NS	NS	A1V1	A2	V1	A2V2	A1	V2
	NS	NS	NS	NS	NS	NS	A2	A2V2	A1V1	V2	V1	A1
	NS	NS	NS	NS	NS	NS	A2V2	V2	A2	A1	A1V1	V1
	NS	NS	NS	NS	NS	NS	V2	A1	A2V2	V1	A2	A1V1

A1 - Zombie scream
A2 - Spider scream
V1 - Zombie
V2 - Spider
A1V1 - combine A1&V1
A2V2 - combine A2&V2
NS - No Stimulus

Figure 3.9: Condizioni

The participant's user number, entered in the "MainMenu" scene, is then passed to the second and final scene named "SampleScene." Within the "SampleScene," an empty object ("AnimationTrigger") with an attached script receives the user number and iterates through the "data_structure" JSON file until it locates a block with the same user number condition.

```
if (PlayerPrefs.HasKey("UserNumber"))
{
    UserNumber = PlayerPrefs.GetString("UserNumber");
    Debug.Log("@" + UserNumber);
}



while (!(ServiceLocator.ExperienceService.CurrentActiveConditions.Select(x => x.Name).ToList().Contains(UserNumber)))
{
    ServiceLocator.ExperienceService.GoToNext<Block>();
}
```

Figure 3.10: Part of the script that selects the pair of blocks

Subsequently, all eight trials from each of the two blocks are executed. Each trial lasts for 2 minutes, during which the stimulus associated with the trial's condition is triggered at the trial's onset. Upon completion of the eighth trial from the second block, the application exits.

3.4 Unity app development

3.4.1 VR scene development

In the construction of the environments for the two scenes constituting the VR experience, a combination of purpose-built 3D models and those made available by the MICT research team from prior experiments were employed. Each model underwent texturing procedures tailored to the requirements of this project, utilizing tools such as Blender and Substance 3D Painter. The creation of frightening stimuli involved the modeling and animation processes within Blender, followed by integration into the Unity platform. The frightening stimuli:

- V1 (visual stimulus number 1) - A 3D model of a zombie that appears in front of the user and translates while screaming into the position of the user's face.

Figure 3.11: V1 frightening stimulus

- V2 (visual stimulus number 2) - A 3D model of a woman appearing in a corner of the kitchen. When she appears, all the lights in the scene turn off, and a spotlight illuminates the woman, who then shifts towards the user's position after 4 seconds.

Figure 3.12: V2 frightening stimulus

- A1 (auditory stimulus number 1) - An empty object with an Audio Source component containing the scream of a zombie.
- A2 (auditory stimulus number 2) - An empty object with an Audio Source component containing the scream of a woman.
- A#V# - Combined auditory and visual stimuli.

3.4.2 Overview of virtual reality experience structure

Figure 3.13: First scene termed "MainMenu"

In the first scene, the user is prompted to enter their participant number (previously communicated to them by the research team). Once the user presses the "Next" button, the second scene is loaded.

Figure 3.14: Second scene termed "SampleScene"

In the second scene, the participant is placed within a virtual kitchen environment, wherein instructional prompts are conveyed via a television display. The objective entails organizing the virtual kitchen by disposing of damaged items (while ensuring proper segregation of waste into appropriate bins) and arranging the remaining items on shelves. After the instructions are communicated to the participant, a 2-minute timer is activated (displayed on the television screen) to allow the participant time to explore the virtual environment. However, during this period, the participant is unable to interact with objects, enabling the collection of physiological data for 2 minutes without any stimuli to establish a baseline. Upon completion of the timer, the experience commences, during which the participant can and must interact with objects to tidy up the kitchen. Throughout this simple task, the participant is subjected to frightening events every two minutes for a total duration of 12 minutes, following which the participant is required to complete a self-assessment questionnaire regarding the frightening level of the experience, which is displayed on the television screen.

Figure 3.15: Shelf and bins of the second scene

Upon completion of the questionnaire, the second and final block of the experience begins. Objects are repositioned on the table to their initial positions, and the participant must repeat the task for another 12 minutes without frightening events. At the conclusion of these 12 minutes, the participant must complete a second questionnaire, after which the application exits, concluding the experience.

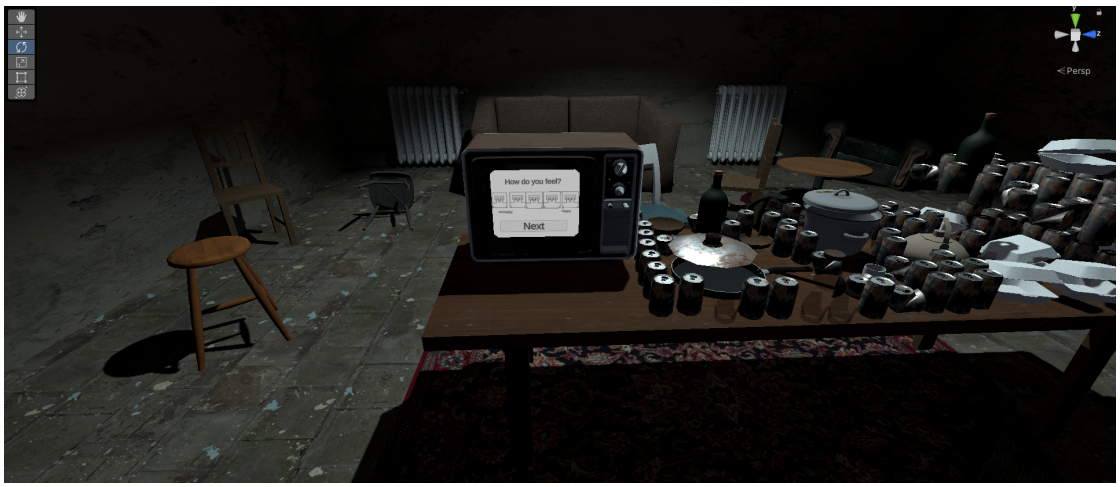

Figure 3.16: Survey displayed on the television

3.4.3 Connection between Vive Focus 3 and Emotibit

As previously explained in section 3.1.2, "OSC Protocol", the Emotibit device communicates with a personal computer via Wi-Fi, utilizing specialized software known as the Emotibit Oscilloscope to manage the data stream. Subsequently, the PC transmits the data through the Wi-Fi network to a predetermined Wi-Fi port, encapsulating the EDA data in OSC messages. These messages, identified by the OSC address "/EmotiBit/0/EDA" contain EDA data formatted as a float

and are transmitted to the designated Wi-Fi port at a rate of 15 times per second. The Unity project utilizes the OSC Jack package to facilitate communication between the Vive Focus 3 VR headset and the PC connected to the Emotibit sensor. The OSC Jack package constitutes a lightweight C# implementation of OSC server/client, primarily designed to offer OSC support within the Unity environment. To establish the connection at the Wi-Fi port within the Unity project, a scriptable object provided by the OSC Jack package is utilized. Within the inspector of this scriptable object, users can specify the IP address and the Wi-Fi port number. Specifically, the default IP address "10.27.91.226" and port "12346" are employed to communicate with an Android device, such as the Vive Focus 3 VR headset.

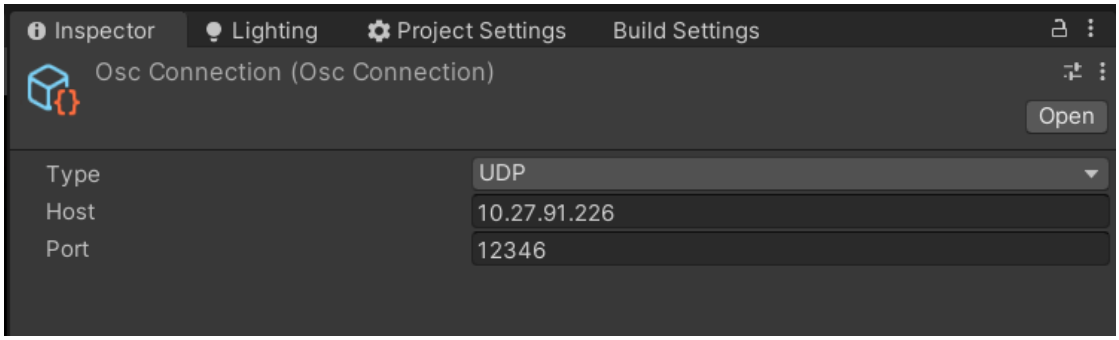

Figure 3.17: Inspector of the scriptable object

Moreover, within the Unity scene, an empty object equipped with the Osc Event Receiver script component (also provided by the OSC Jack package) is employed. Within the inspector of this empty object the following parameters are defined:

- The OSC address of the intended OSC message to be read.
- The data type of the information contained within the OSC message.
- The method to be invoked each time the receiver detects an OSC message with the specified address.

The method invoked by the Unity event, entailing the reading of an OSC message with the address "/EmotiBit/0/EDA" belongs to the script named "EdaWriter". This method stores the value in a List and performs signal operations (which will be elaborated upon in Section 3.4.3, titled "Pre-processing Electrodermal Activity Algorithm").

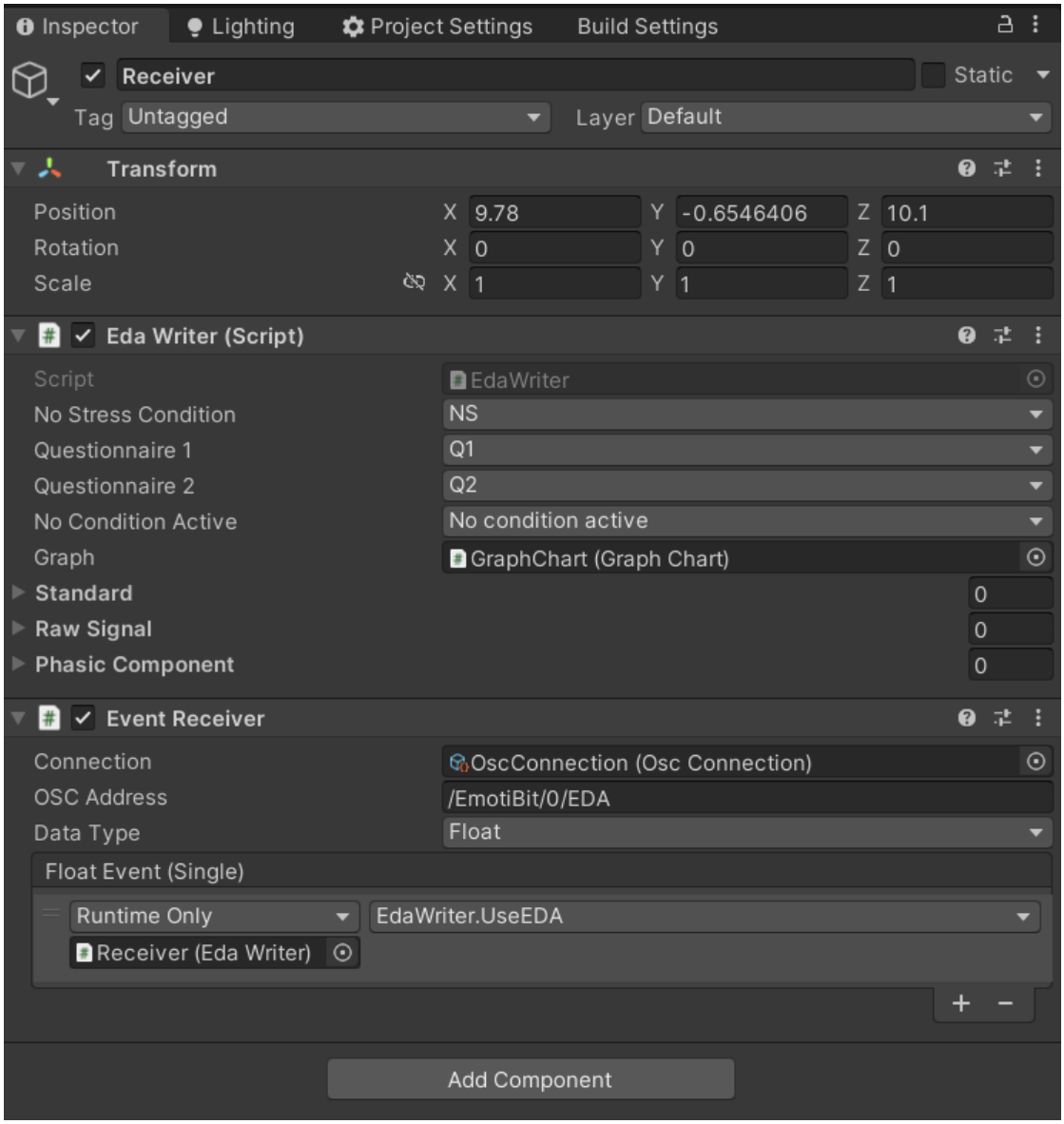

Figure 3.18: Inspector of the empty object equipped with the OSC Event Receiver script component

3.4.4 Pre-processing electrodermal activity algorithm

As previously introduced in the subsection 3.1.1 "EDA signal", the electrodermal activity signal comprises two distinct components: the tonic and phasic components. The tonic component represents the baseline or general level of EDA activity, encompassing slower-acting characteristics and background variations in the signal,

such as overall level, gradual changes, and sustained trends over time. Skin Conductance Level (SCL) is a commonly used measure to quantify the tonic component, with alterations in SCL reflecting general fluctuations in autonomic arousal levels. In contrast, the phasic component of EDA pertains to the more rapid and transient fluctuations in the signal, often referred to as Skin Conductance Responses (SCRs). These responses capture the dynamic changes in electrodermal activity that occur in response to specific stimuli or events. Research suggests that both tonic and phasic components play significant roles in reflecting autonomic arousal, with distinct neural mechanisms underlying each component.

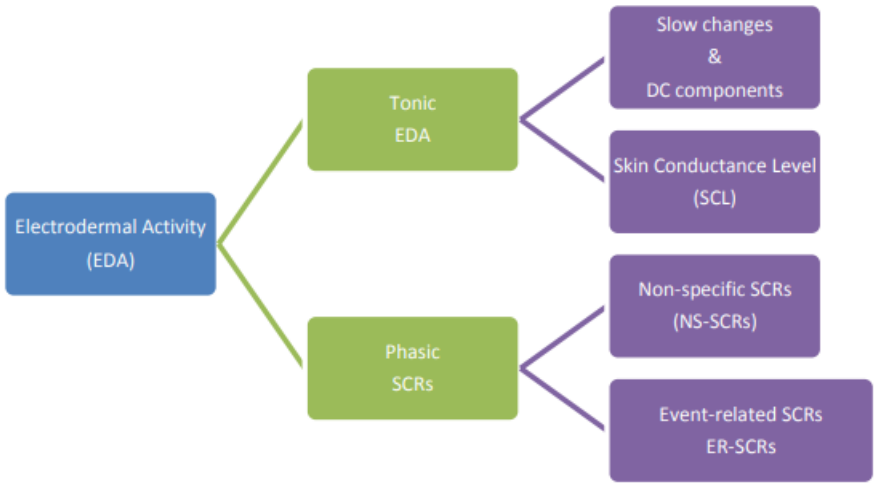

Figure 3.19: Components of the EDA signal

A common procedure performed on EDA data is standardization. Standardization involves adjusting the data to enable direct and meaningful comparisons between individuals. While standardization may not always be essential, it is challenging to envision a scenario where transforming any Skin Conductance Responses (SCRs) from an individual into a coordinate system reflecting the constraints of their responsiveness would not be advisable. The objective of standardization is to streamline comparisons of individual differences, ensuring that all SCRs are adjusted based on their physiological responsiveness. This process helps mitigate factors such as skin thickness, facilitating more accurate comparisons across individuals. [13]

Python pre-processing algorithm executed on a server

As mentioned in subsection 3.3, the Unity project developed for the MICT research center utilizes the ExperienceDNA Framework to process real-time data received from a server. This data includes physiological signals, such as Electrodermal Activity (EDA), transmitted by the VR headset running the application. The preprocessing algorithm for EDA, executed by the server, is written in Python and employs the NeuroKit2 library, which offers convenient access to advanced biosignal processing routines. The Python script conducts a comprehensive analysis of the EDA signal, encompassing data loading, cleaning, processing, visualization, decomposition, and skin conductance and power spectral density analysis. Specifically, the operations performed by the script include:

Loading and cleaning data:

- Loading the EDA data from a JSON file into a Pandas DataFrame.
- Converting Unity timestamps to Unix time.
- Removing rows with missing or invalid values.

Signal processing:

- Processing the raw EDA signal to derive raw data, tonic response, phasic response, SCR onset, and SCR peak using the NeuroKit2 library.
- Cleaning the EDA signal.
- Extracting features such as SCR onsets, SCR peaks, SCR recovery, and SCR amplitude.
- Visualizing the computed information.

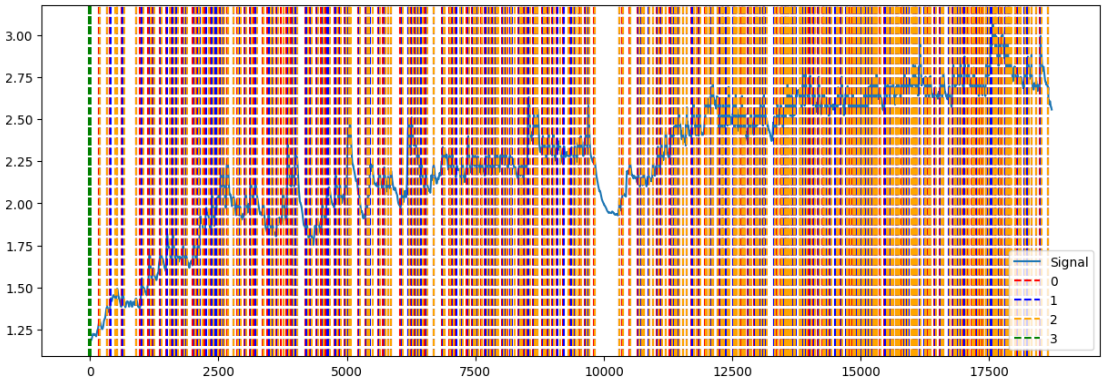

Figure 3.20: Output of the "Signal Processing" part of the Python script

Decomposition:

- Decomposing the EDA signal into phasic and tonic components using NeuroKit2.
- Extracting the phasic and tonic components from the decomposition.
- Visualizing the extracted information.

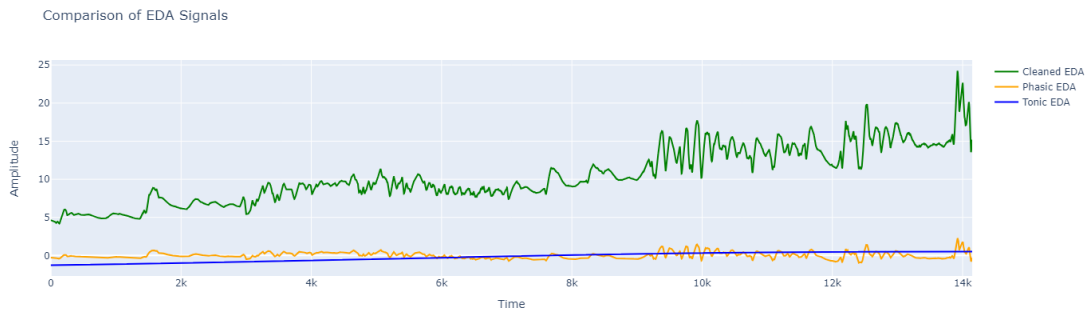

Figure 3.21: Output of the "Decomposition" part of the Python script

Power spectral density (PSD) analysis:

- Calculating the PSD using Welch's method to estimate the frequency with the highest power in the EDA signal.
- Obtaining the highest density interval using random samples.
- Visualizing the computed information.

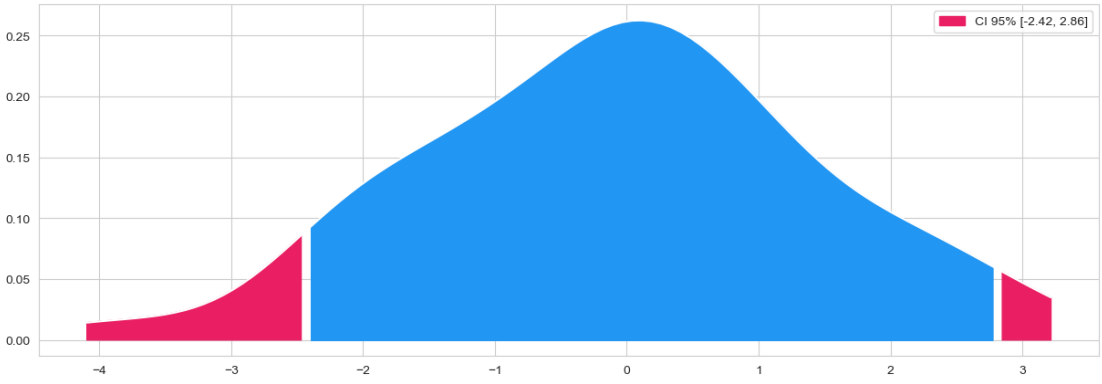

Figure 3.22: First output of the "Power spectral density (PSD) analysis" part of the Python script

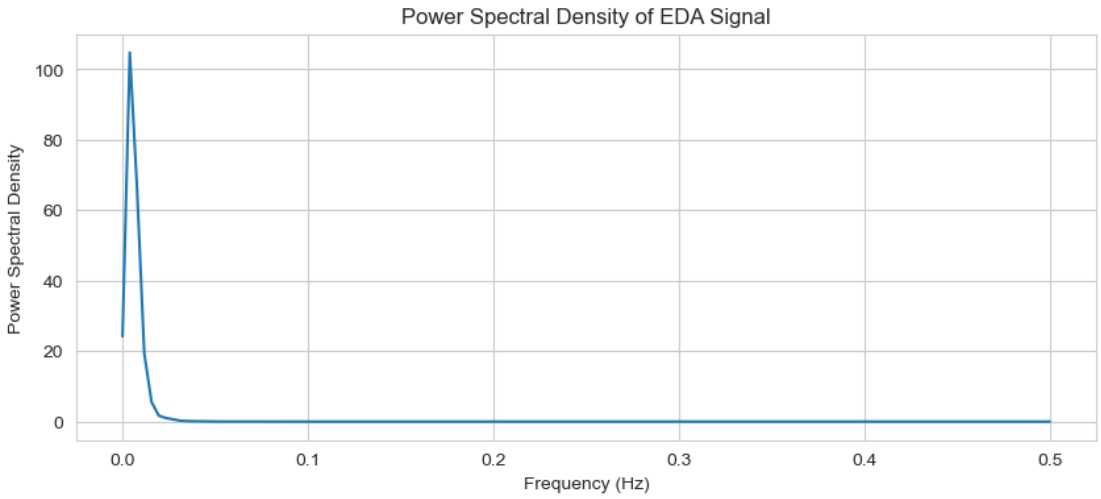

Figure 3.23: Second output of the "Power spectral density (PSD) analysis" part of the Python script

For this thesis project, only the Python algorithm's functionality responsible for cleaning and extracting the fundamental components of the signal (phasic and tonic) is considered. Thus, only the part of the script defined above as "Decomposition" is considered. The C# algorithm, operating in real-time within the Unity application, solely extracts the phasic and tonic components. Now, let's delve more into the part of the Python script referred to as "Decomposition." The initial signal processing operation involves the "eda_clean()" method from the NeuroKit2 library.

```
cleaned_eda = nk.eda_clean(df['Value'], sampling_rate=15, method='neurokit')
```

Figure 3.24: eda_clean() method

This method discards all floats outside the range of 0.03 to 30 and all floats equal to NaN. Subsequently, the signal undergoes standardization. As previously explained, standardization facilitates the comparison of EDAs from different users by reducing error margins associated with individual user characteristics. Finally, the standardized signal is passed as an argument to a method called "eda_phasic()" which decomposes the signal into its two fundamental components: tonic and phasic. This latter method, provided by the NeuroKit2 library, facilitates the decomposition of Electrodermal Activity (EDA) signals into two distinct components: phasic and tonic. Here is a summary of the methods and functionalities provided by the "eda_phasic()" function:

- High-pass filtering: this method employs a high-pass filter with a default cutoff

```
# Decompose the EDA signal into phasic and tonic components
eda_components = nk.eda_phasic(nk.standardize(cleaned_eda))


# Get the phasic and tonic components
phasic_component = eda_components["EDA_Phasic"]
tonic_component = eda_components["EDA_Tonic"]
```

Figure 3.25: Decomposition of the EDA signal into the two components

frequency of 0.05 Hz. Users have the flexibility to adjust the cutoff frequency using the "cutoff" argument.

- Median smoothing: by employing a median value smoothing filter, this method effectively eliminates regions of rapid change in the raw EDA signal. The Phasic component is derived by subtracting the smoothed signal from the original. The processing duration is contingent upon the smoothing factor, regulated by the "smoothing_factor" argument.

- cvxEDA: this approach adopts convex optimization principles for EDA processing and necessitates the installation of the cvxopt package.

- SparsEDA: this method employs sparse non-negative deconvolution techniques.

The function encompasses the following parameters:

- "eda_signal": represents the raw EDA signal.

- "sampling_rate": denotes the sampling frequency of the raw EDA signal in Hz.

- "method": specifies the processing pipeline to be applied, including options such as "cvxEDA", "smoothmedian", or "highpass".

- "kwargs": facilitates the provision of additional arguments to the specific method.

Upon execution, the function yields a DataFrame containing the tonic and phasic components as distinct columns.

C# pre-processing algorithm executed by the VR headset

As elucidated in subsection 3.4.3, "Connection between Vive Focus 3 and Emotibit," the reading of an OSC message on the Wi-Fi port "12346" and IP address "10.27.91.226" triggers a Unity event invoking the "useEDA" method of the EdaWriter script. This method accepts a float argument, representing the information from the received OSC message, which is discarded if it falls outside the range of 0.03 – 30 or if it equals NaN. If not discarded, the float is stored in a List<float> object named "CleanedSignal." The signal, cleansed of any artifacts, is visualized using a chart visualization package called "Chart and Graph." Upon each new sample being stored, the entire list of values undergoes standardization.

Signal Standardization For each new value recorded, the total signal mean is recalculated using the "Mean()" method along with the standard deviation.

```
public List<double> Standardization(List<float> samples)
{
    List<double> StandardizeSignal = new List<double> { };
    for (int i = 0; i < samples.Count; i++)
    {
        StandardizeSignal.Add((samples[i] - Mean(samples)) / StandardDeviation(samples));
    }
    return StandardizeSignal;
}
```

Figure 3.26: Standardization method

The standard deviation (σ) involves the summation of squared differences between each signal value and the signal mean. This summation is divided by the total number of signal values, and the square root of this division is taken.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N}}$$

Where:

- σ is the standard deviation,
- x_i represents each individual data point in the dataset,
- μ is the mean (average) of the dataset,
- N is the total number of data points in the dataset.

Phasic component The EdaWriter script, which contains the UseEDA method (invoked with each Osc message read), utilizes a namespace called "DSP" that enables the creation of an object of type "LowpassFilterButterworthImplementation," referred to as "lowpass," representing a Butterworth filter with a cutoff frequency of 3 Hz and fourth order. The "lowpass" object encompasses the compute method,

```
public float StandardDeviation(List<float> samples)
{
    float sommatoria = 0f;


    if (samples.Count == 0)
    {
        Debug.LogError("Sample list is empty!");
        return 0f; // Return 0 if the list is empty
    }

    foreach (float edaFrame in samples)
    {

        sommatoria += Mathf.Pow(edaFrame - Mean(samples), 2f);
    }


    return Mathf.Sqrt(sommatoria / (samples.Count - 1));
}
```

Figure 3.27: Standard deviation method

```
private LowpassFilterButterworthImplementation lowpass = new LowpassFilterButterworthImplementation(3, 4, 15);
```

Figure 3.28: Low-pass Butterworth filter object

which operates on lists of floats similar to how a filter operates on a signal. Thus, the standardized signal undergoes processing through the compute method, and the final output of the compute method is the phasic component. Specifically, the DSP namespace simulates a Butterworth low-pass filter with the following characteristics: it is designed to allow the passage of frequencies below a designated cutoff frequency while attenuating frequencies above that value. The principal features of a Butterworth low-pass filter include a smooth frequency response and a rapid attenuation beyond the cutoff frequency. Here are the typical features of a Butterworth low-pass filter:

- Frequency Response: The Butterworth low-pass filter is designed to have a flat frequency response within the passband, without oscillations or undesired peaks. This means that frequencies below the cutoff frequency pass through the filter with minimal attenuation, while higher frequencies are significantly attenuated.

- Attenuation Roll-off: After the cutoff frequency, the attenuation of the Butterworth low-pass filter drops rapidly, ensuring a clear separation between the passband and the stopband.

- Filter Poles: The poles of the Butterworth filter are positioned on the unit circle in the complex plane to achieve the desired frequency response. This pole placement contributes to the characteristic of maximum flatness in the frequency response.

- Filter Order: The order of the Butterworth filter determines the slope of the transition between the passband and the stopband. A higher filter order results in a steeper transition slope.

[14]

Tonic component Regarding the Tonic component, it awaits filling a buffer of 4000 samples before commencing its calculation. This component is determined by applying a Median Filter to the standardized signal, where the median filter operates within a window comprising 4000 values.The median filter distinguishes itself from conventional numerical filters such as the moving average and Savitzky-Golay filters due to its effectiveness in managing spikes and anomalies in the data while preserving the signal's general shape. Whereas the moving average tends to smooth the signal and Savitzky-Golay filters are more suited for curve approximation, the median filter excels in removing isolated spikes without compromising signal resolution. It is particularly adept at handling "spiky" noise or sudden spikes in the data, facilitating the clear separation of peaks from the slowly evolving baseline, which is crucial for data interpretation in analytical contexts. The median filter operates on the signal through the following steps:

- It arranges the values within a designated moving window.
- It identifies the median value, which represents the middle value when the values are arranged in ascending or descending order.
- It substitutes the original value with the median value within the moving window.
- It iterates this process for each point in the signal, moving the moving window along the signal.

Essentially, the median filter replaces each value in the signal with the median value of the surrounding values, aiding in spike elimination and noise reduction without compromising the overall signal shape, thus yielding the Tonic component of the EDA. [15]

```
private static double[] MedianFilter(double[] signal, int size)
{
    var smoothed = new double[signal.Length];
    for (int i = 0; i < signal.Length; i++)
    {
        int startIndex = Mathf.Max(0, i - size / 2);
        int endIndex = Mathf.Min(signal.Length - 1, i + size / 2);
        double[] subset = new double[endIndex - startIndex + 1];
        Array.Copy(signal, startIndex, subset, 0, endIndex - startIndex + 1);
        Array.Sort(subset);
        if (subset.Length % 2 == 0)
        {
            smoothed[i] = (subset[subset.Length / 2] + subset[subset.Length / 2 - 1]) / 2.0;
        }
        else
        {
            smoothed[i] = subset[subset.Length / 2];
        }
    }
    return smoothed;
}
```

Figure 3.29: Median filter method

Visualization Upon registering a new value of the cleaned EDA signal and subsequently recalculating the phasic component, two methods are invoked to visualize the data on a graph overlay located in the bottom-right corner of the user's view. After recalculation, all points on the phasic component graph are cleared using the "ClearCategory" method, and the new values are added using a for loop with the "AddPointToCategory" method.

```
Graph.DataSource.ClearCategory("PhasicComponent");
for (int i = 0; i < Standard.Count; i++)
{
    Graph.DataSource.AddPointToCategory("PhasicComponent", i, PhasicComponent[i], 1f);
```

Figure 3.30: Script that visualizes the phasic component

Regarding the cleaned EDA, the "AddPointToCategory" method is simply utilized to append the new sample to the graph. As for the tonic component (since

```
Graph.DataSource.ClearCategory("PhasicComponent");
for (int i = 0; i < Standard.Count; i++)
{
    Graph.DataSource.AddPointToCategory("PhasicComponent", i, PhasicComponent[i], 1f);
```

Figure 3.31: Script that visualizes the EDA cleaned signal

its calculation depends on a window of 4000 values, as explained earlier), the values are recalculated every 4000 values. Thus, for every 4000 recorded values, the points on the tonic component graph are removed, and the newly computed values are added again.

Figure 3.32: Overlay EDA Graph

Chapter 4

Comparative analysis of Python and C# algorithms

This chapter presents a comprehensive comparative analysis of algorithms implemented in Python and C#. The overarching goal is to meticulously evaluate the congruence between the output data generated by both algorithms, with the Python implementation serving as the gold standard reference. The imperative to compare the Python and C# algorithms stems from the critical need to ensure methodological consistency and data processing accuracy. Originating from the MICT team of Researchers specialized in physiological data analysis, the Python algorithm represents a well-established benchmark against which the C# implementation is rigorously evaluated. The investigation is driven by the necessity to validate the equivalence of the C# algorithm in relation to this esteemed standard. To fulfill the research objectives, a meticulous blend of qualitative and quantitative analyses is employed. Central to the evaluation is the assessment of data congruence, achieved through the computation of the mean square error (MSE) and correlation coefficient. These statistical metrics serve as pivotal indicators of the similarity between the output datasets generated by the Python and C# algorithms. These analyses were conducted concerning 2 vectors of floats, as both the Python pre-processing algorithm (employed by the MICT research team) and the C# algorithm crafted explicitly for this thesis project generate 2 vectors of floats representing: the phasic component of the EDA, and the tonic component of the EDA. The raw electrodermal activity data, unprocessed by the algorithms, were furnished by the MICT research team, who collected these data using the virtual reality horror application integrated with Emotibit. Subsequently, the team processed the data using the Python algorithm, albeit not in real-time. Consequently, the raw data provided by MICT will now undergo processing by the C# algorithm and subsequently be compared with the same data processed by the Python algorithm (also provided

by MICT). The successful validation of the C# algorithm not only underscores its technical proficiency but also holds significant implications for real-time integration within the VR environment of the Unity application. Such integration would empower dynamic adjustments based on users' emotional states, enhancing the immersive experience. The chapter's structural framework is thoughtfully designed to accommodate diverse analyses and tests aimed at comprehensively comparing the Python and C# algorithms. Each section delves into specific methodologies and unveils findings pertinent to the assessment of data congruence.

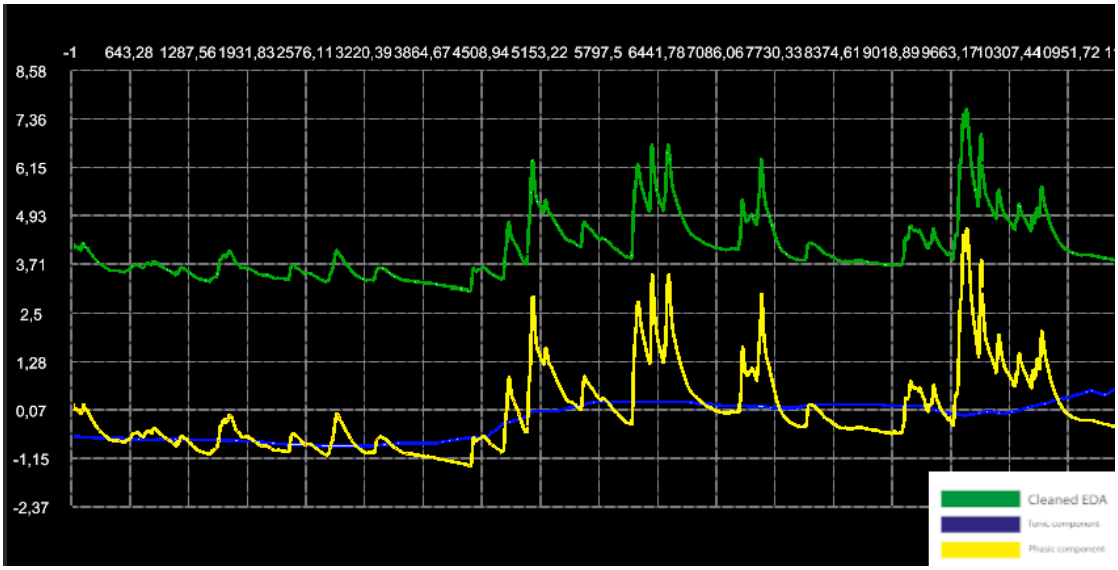

Figure 4.1: EDA graph of the first dataset elaborated by the C# algorithm

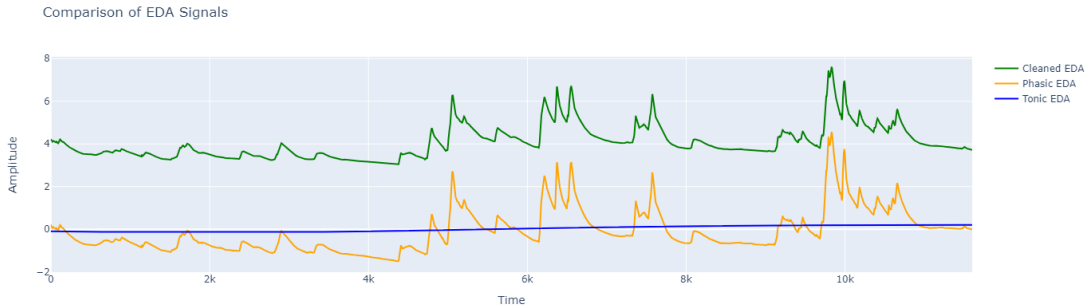

Figure 4.2: EDA graph of the first dataset elaborated by the Python algorithm

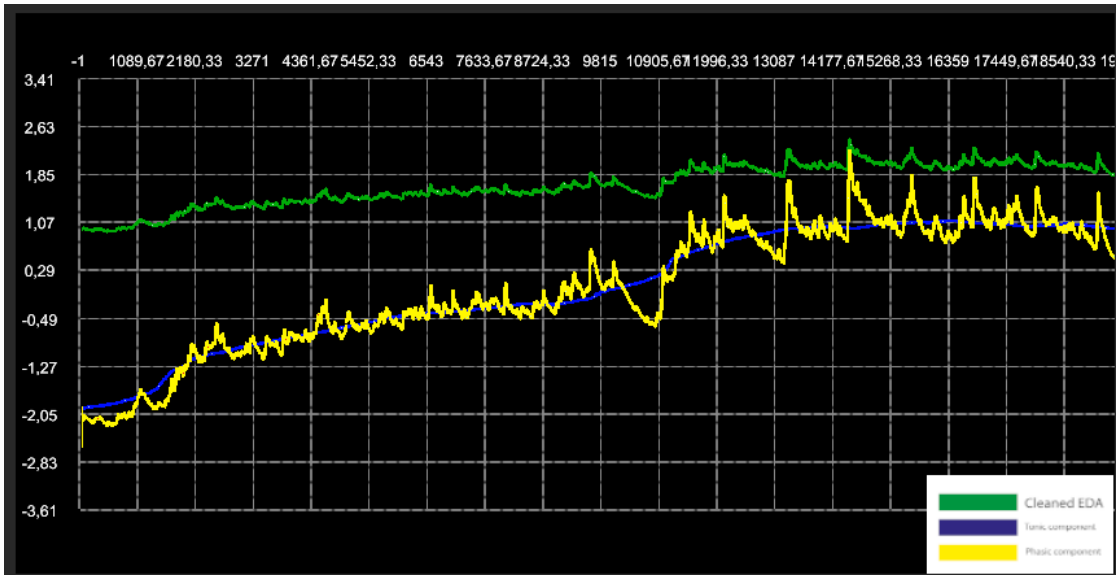

Figure 4.3: EDA graph of the thirteenth dataset elaborated by the C# algorithm

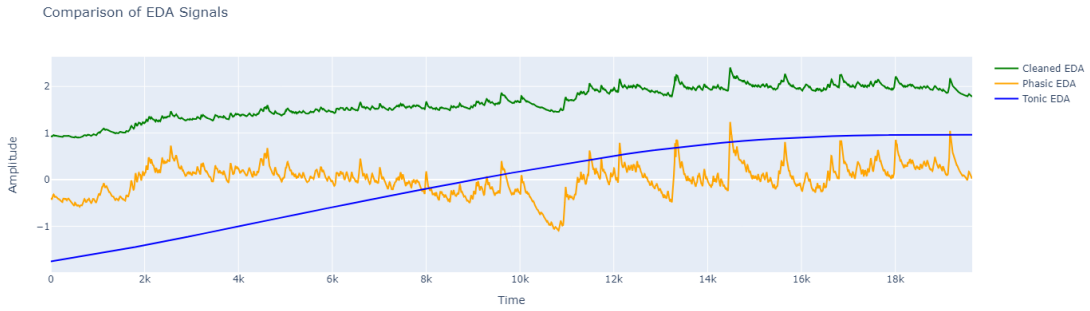

Figure 4.4: EDA graph of the thirteenth dataset elaborated by the Python algorithm

4.1 Mean square error analysis

To confront the output datasets yielded by the two EDA pre-processing algorithms, we computed the mean square error. The Mean Square Error (MSE) is a commonly used metric for assessing the discrepancy between predicted values (dataset obtained as the output of the Python algorithm) and observed values (dataset obtained as the output of the C# algorithm) within a dataset. The formula for MSE is given by:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

where:

- n is the total number of samples in the data,
- Y_i represents the observed values,
- $\hat{Y}_i$ represents the predicted values.

MSE computes the average of the squared differences between the predicted and observed values. It serves as a common measure of accuracy and is frequently employed in statistics, signal processing, and other fields to assess the goodness of fit of a model to the observed data[16]. The MSE was computed across sixteen datasets. Specifically, the MICT research team provided sixteen JSON files representing raw Electrodermal Activity data for this project. These sixteen JSON files contain a series of floating-point numbers and were recorded and generated by the Emotibit sensor during sixteen experiments, where sixteen different participants were exposed to 16 distinct sequences of alarming stimuli inside the VR horror experience. Each sample recorded in the files was acquired at a sampling frequency of 16 Hz by the Emotibit sensor. For each raw EDA dataset, both the phasic and tonic components were computed using both the Python algorithm (the established standard) and the C# algorithm (the algorithm under evaluation). The following table illustrates the variation in MSE for the two tonic and phasic components across different datasets:

Based on the computed MSE values for both the phasic and tonic components across the 16 datasets, several observations can be made regarding the performance and characteristics of the algorithms. Firstly, it is evident that the MSE values for the tonic component tend to be consistently lower compared to those of the phasic component across most datasets. This suggests that the algorithm, particularly in its treatment of the tonic aspect of electrodermal activity (EDA), is generally more accurate and closer to the expected values derived from the Python algorithm, which serves as the reference standard. The variability in MSE values across different datasets highlights the sensitivity of the algorithms to variations in input data. Datasets with higher MSE values may indicate instances where the algorithms struggled to accurately capture the underlying patterns in the EDA data, potentially due to factors such as noise, artifacts, or other sources of variability inherent in the physiological signals. Moreover, the MSE values for both components exhibit fluctuations across different datasets, indicating that the performance of the algorithms is not uniform and may vary depending on the specific characteristics of the data being processed. This underscores the importance

Mean square error		
Dataset Number	MSE tonic component	MSE phasic component
1	0.173451140274	0.037826347018
2	0.194380000302	0.580371393254
3	0.056288442856	0.716928713647
4	0.057521483993	0.993225514158
5	0.076703859630	0.737314709473
6	0.028382118136	0.848453762497
7	0.112478235421	0.399060974230
8	0.096911427922	0.825341247253
9	0.050617437290	0.886302221942
10	0.117698627982	0.744071584296
11	0.034911732766	0.916510425580
12	0.028783787445	0.803057469811
13	0.046710181064	0.845920446125
14	0.167427308818	0.895542753969
15	0.008191999655	0.875872674268
16	0.042824566898	0.757753799573

Table 4.1: Mean square error table

of robustness and adaptability in algorithm design, particularly in the context of real-world applications where data quality and conditions may vary.

In summary, the analysis of MSE values provides valuable insights into the performance and accuracy of the algorithms in processing EDA data. The observed differences between the MSE values for the phasic and tonic components underscore the complexity of EDA signal processing and highlight areas for potential algorithm refinement and optimization. Further investigation and refinement of the algorithms may be necessary to enhance their accuracy and robustness across diverse datasets and conditions. However, the tonic and phasic components play crucial roles in interpreting variations in electrodermal activity over both long and short periods. Therefore, the absolute value of the components is not as significant as the shape of the graph depicting the variations. In the upcoming subsection, the correlation coefficient between the output data generated by the two algorithms will be analyzed. This coefficient enables the quantification of the degree of agreement in the observed changes between the two outputs, providing a measure of consistency in the results across the algorithms.

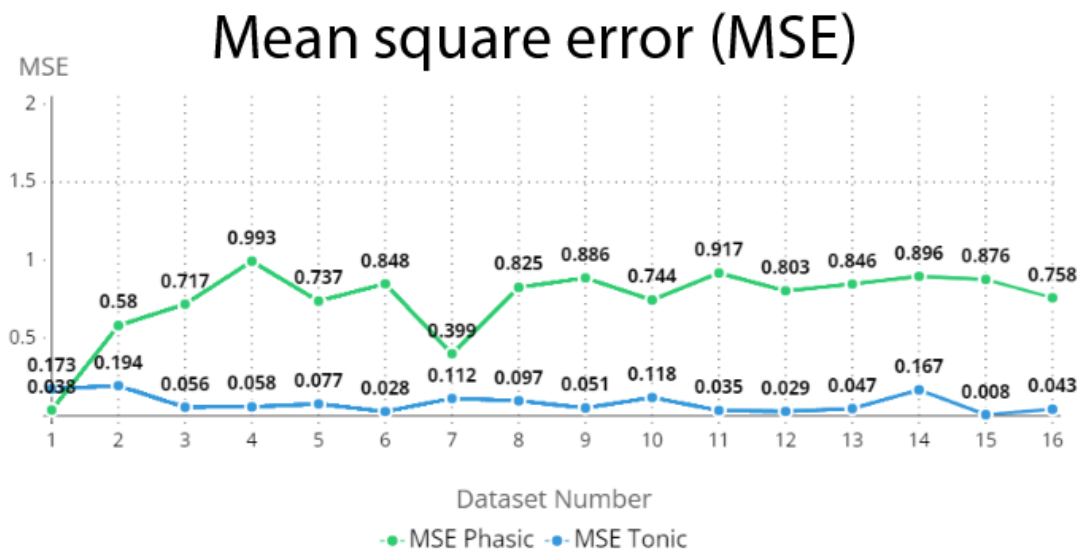

Figure 4.5: MSE graph of the tonic and phasic component

4.2 Correlation analysis

The correlation coefficient is a statistical measure indicating the strength and direction of the relationship between two variables. In other words, it reflects how much the two variables tend to vary together. There are different types of correlation coefficients, but one of the most common is the Pearson correlation coefficient, denoted by the symbol r. The Pearson correlation coefficient is calculated using the following formula:

$$r = \frac{\sum(X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum(X - \bar{X})^2 \sum(Y - \bar{Y})^2}}$$

where:

- X and Y are the variables being correlated.
- $\bar{X}$ and $\bar{Y}$ are the means of variables X and Y.
- $\sum$ denotes the summation over all variable values.
- $(X - \bar{X})$ and $(Y - \bar{Y})$ are the deviations from the means of X and Y, respectively.

The Pearson correlation coefficient can range from -1 to 1:

- If $r = 1$, the variables are perfectly positively correlated.
- If $r = -1$, the variables are perfectly negatively correlated.
- If $r = 0$, there is no linear correlation between the variables.

Generally, a value closer to 1 or -1 indicates a stronger correlation, while a value close to 0 indicates a weaker correlation. The correlation coefficient (CC) serves as a valuable metric for evaluating the relationship between two datasets. In the context of our analysis, the CC provides insights into how closely the output variations of the Python and C# algorithms align.[17]

Correlation coefficient (CC)		
Dataset Number	CC tonic component	CC phasic component
1	0.892175501687	0.985234303901
2	0.897073896366	0.656760256008
3	0.979884842830	0.563975136299
4	0.984637862367	0.486277584494
5	0.940708319121	0.538005741214
6	0.993293937328	0.404851357344
7	0.862266661971	0.775906002708
8	0.974507322798	0.478315701349
9	0.983966295329	0.346493020805
10	0.939364462617	0.506529312402
11	0.99138673657	0.407549102801
12	0.985070427563	0.466895235713
13	0.98033747677	0.411242060632
14	0.90937042158	0.339958788262
15	0.99760319925	0.480921722882
16	0.984728753418	0.501096416377

Table 4.2: Correlation coefficient table

Examining the CC values calculated for both the phasic and tonic components across the 16 datasets reveals notable trends. Specifically, the CC values for the tonic component consistently approach or surpass 0.9, indicating a strong positive correlation between the outputs of the two algorithms. This suggests that variations in the tonic component exhibit a high degree of similarity between the Python and C# algorithms across different datasets. Conversely, the CC values for the phasic component demonstrate more variability, with some datasets exhibiting moderate correlations while others show weaker correlations. Despite this variability, certain

datasets still display CC values approaching 0.9, indicating a notable level of alignment between the phasic outputs of the two algorithms in those instances.

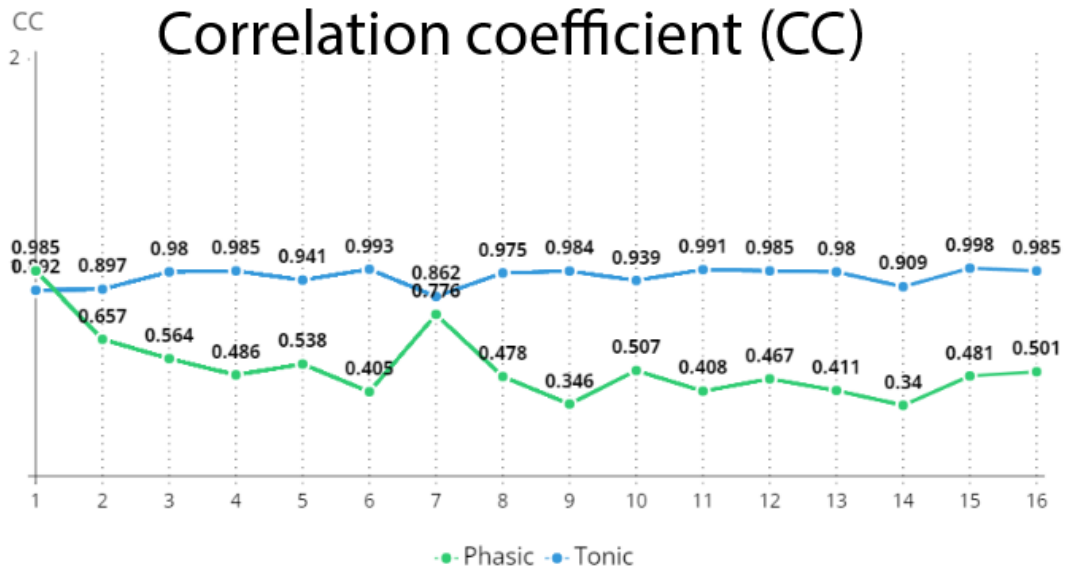

Figure 4.6: CC graph of the tonic and phasic component

Chapter 5

Analysis of EDA measurements

In this subsection, an in-depth analysis of EDA data collected from 16 experiments is conducted. Each experiment entails the measurement of EDA using the Emotibit sensor, capturing physiological responses to distinct sequences of frightening stimuli. These stimuli are carefully arranged to induce frightening events, thus eliciting varying degrees of arousal and emotional reactions from participants. The datasets, processed by the Python pre-processing algorithm, are analyzed and compared to detect peaks and assess variations in EDA over short and long periods. Through systematic analysis, this section aims to elucidate patterns, trends, and correlations within the EDA data, providing valuable insights into the psychophysiological responses triggered by different frightening stimuli sequences. Additionally, each dataset is accompanied by a self-assessment questionnaire, providing further context for the analysis and interpretation of the EDA data.

Structure of the self-assessment questionnaire

The questionnaire is presented on the television screen at the end of each block within the virtual environment. The initial screen prompts users to self-assess their levels of happiness, calmness, and perceived control on a scale ranging from 1 to 10.

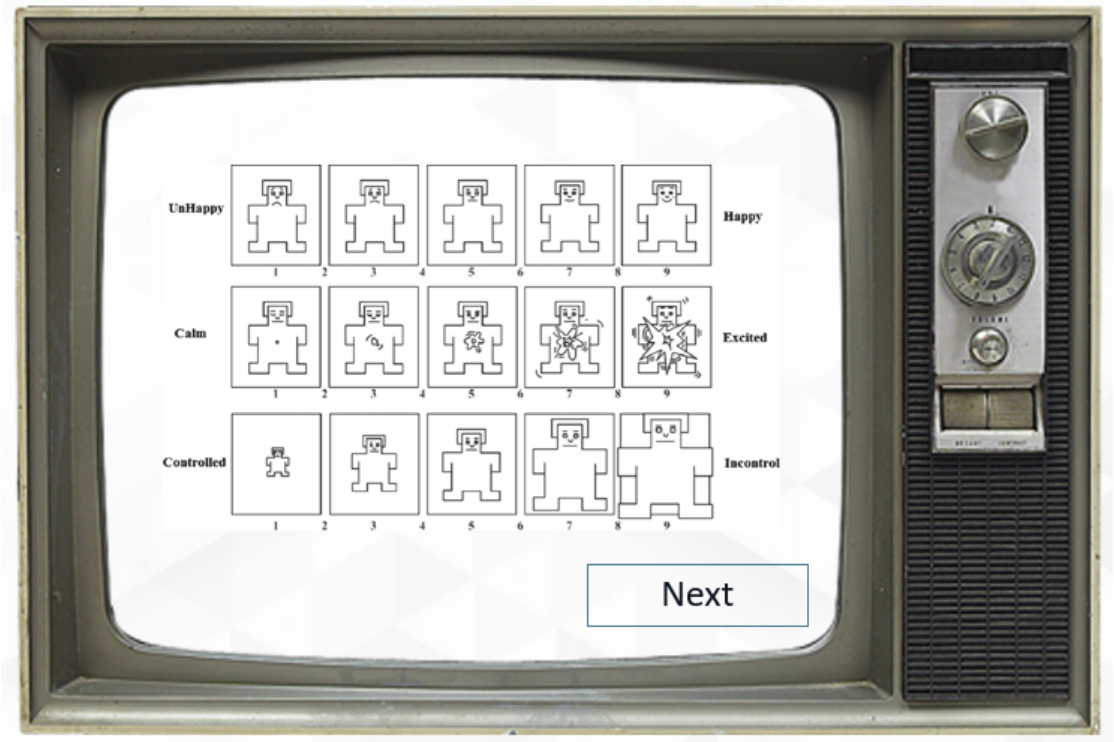

Figure 5.1: Self-Assessment Manikin

Additionally, users are asked to rate their emotional engagement with the virtual environment on a scale from 1 to 10, where 1 indicates "Completely disagree" and 10 indicates "Completely agree" in response to statements regarding the sense of presence within the VR experience and the intensity of frightening events. This questionnaire aims to gather data on users' perceptions of stimulation and immersion within the virtual environment throughout the experience. Such feedback is crucial for evaluating the effectiveness of the virtual experience, assessing user engagement, and collecting insights that can inform future developments in virtual reality technology. Furthermore, it is crucial to ascertain whether the peaks and variations in EDA are attributable to fear or other emotional states of the user.

Figure 5.2: Second part of the questionnaire

EDA data

The 16 datasets have been categorized based on their associated questionnaires. All responses are on a scale of 1 to 10, where 1 denotes minimal involvement and a very low level of fear or stimulation, while 10 indicates maximum involvement and a very high level of fear and engagement in the application. For each experiment or dataset, the average of responses, ranging from 1 to 10, was calculated. Thus, each dataset is assigned a value representing the participant's perceived involvement and fear. The datasets were then divided into three groups based on the average responses to their respective questionnaires. Division of groups:

- Group 1: datasets with an average response below 3.33
- Group 2: datasets with an average response between 3.33 and 6.66
- Group 3: datasets with an average response above 6.66

Group 1	Group 2	Group 3
Dataset 2 Dataset16	Dataset4	Dataset1 Dataset3 Dataset5 Dataset6 Dataset7 Dataset8 Dataset9 Dataset10 Dataset11 Dataset12 Dataset13 Dataset14 Dataset15

Table 5.1: Dataset division in groups

As observed from the graph, the majority of questionnaires exhibit an average response above 6.66. In these datasets, pronounced peaks in the phasic component of EDA are noticeable during frightening events. Generally, it can be affirmed that variations in the phasic component of EDA occur exclusively during audiovisual and auditory events, thus signaling that purely visual events have lesser efficacy. Regarding the tonic component of EDA, variations over a long period are evident only in datasets with an average response exceeding 3.33, namely those in Group 2 and Group 3.

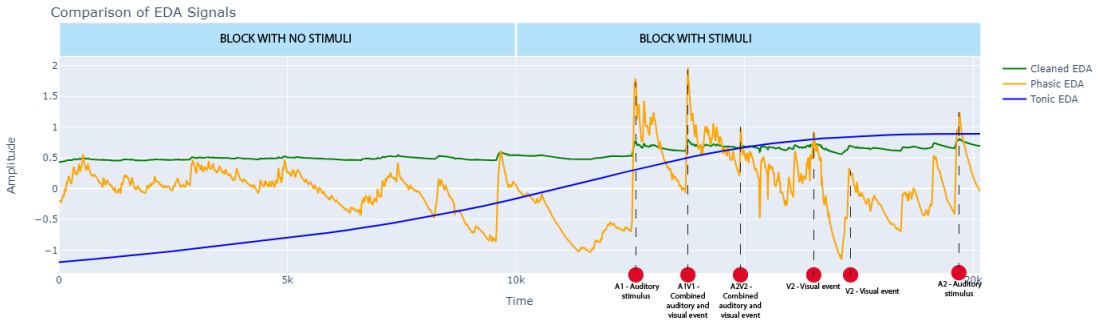

Figure 5.3: EDA graph of the tenth dataset elaborated by the Python algorithm

The tonic component of EDA, indicating a long-term change in emotional state, exhibits a variation between the first and second blocks of the experiment in all datasets of the second and third groups. Specifically, it is low in the first

block (devoid of frightening events) and higher in the second (which includes six frightening events).

Figure 5.4: EDA graph of the forth dataset elaborated by the Python algorithm

Indeed, as depicted in the graph above, in dataset number 4 (which belongs to Group 2, hence exhibiting an average perceived fear), there are no remarkably high peaks corresponding to frightening events (in terms of the phasic component). However, the tonic component still indicates an increase in EDA over the long term with the onset of the second block of the experiment (containing frightening stimuli). Additionally, it can be observed that the event with the highest peak (in the phasic EDA component) is within the eleventh dataset, which, precisely, is within Group 3 (the group with datasets whose participants declared to have been particularly frightened and engaged by the experience).

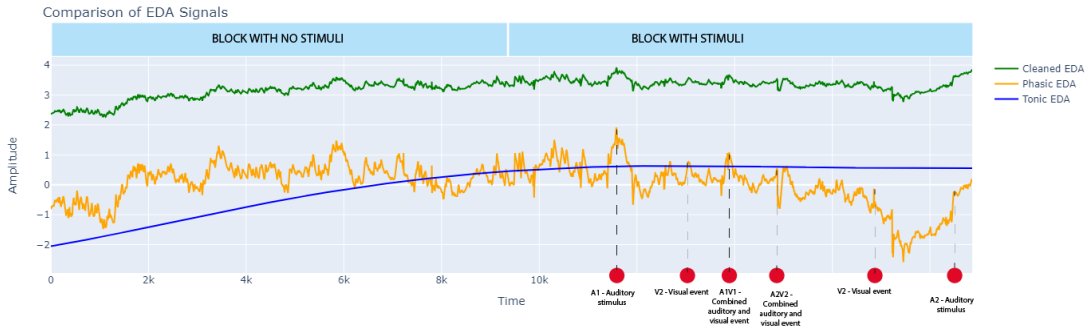

Figure 5.5: EDA graph of the fifth dataset elaborated by the Python algorithm

In conclusion, the alignment between self-assessment questionnaires and EDA data suggests, albeit with a limited dataset sample, that EDA could be a reasonably reliable indicator for assessing the level of fear experienced by a user.

Chapter 6

Results and possible future developments

In conclusion, the convergence between self-assessment questionnaires and EDA data, albeit within the constraints of a limited dataset sample, suggests the potential of EDA as a reliable indicator for evaluating user fear levels in virtual reality experiences. This alignment underscores the utility of physiological measurements in augmenting subjective assessments, thereby enhancing our understanding of user engagement and emotional responses within immersive environments. Notably, the observed predominance of pronounced EDA peaks during frightening events, particularly those of auditory nature, suggests a heightened physiological response compared to purely visual stimuli. This underscores the differential impact of sensory modalities on emotional arousal and highlights the importance of considering multimodal stimuli in immersive experiences. However, further research with larger sample sizes and refined methodologies is warranted to validate and extend these preliminary findings, providing deeper insights into the interplay between sensory stimuli, physiological responses, and subjective experiences in virtual reality environments. The ability to accurately capture and analyze physiological responses, particularly in dynamic environments such as VR experiences, holds significant implications for various domains including healthcare, entertainment, and human-computer interaction. Looking ahead, there are compelling opportunities for future development and refinement of the application. One promising avenue involves leveraging the insights gained from the analysis to enhance the VR horror experience. By integrating real-time monitoring of users' frightening levels and physiological responses, the VR environment can be dynamically adapted to modulate the intensity and timing of frightening stimuli. This adaptive approach not only enhances the immersive quality of the experience but also ensures a personalized and engaging encounter for each user. Furthermore, exploring strategies to mitigate

habituation effects and sustain user engagement represents another exciting frontier. By introducing variability in the sequences of audiovisual stimuli and dynamically adjusting the pacing and intensity of scares, effective prolongation of suspense and maintenance of users' emotional arousal throughout the experience can be achieved. This proactive approach to content delivery not only enriches the overall user experience but also opens new avenues for research and innovation in VR entertainment and psychological interventions. In conclusion, this study underscores the potential of EDA data processing algorithms to revolutionize immersive experiences in virtual environments. By harnessing the power of data-driven insights and emerging technologies, new possibilities for engaging, impactful, and emotionally resonant experiences in the realm of virtual reality are poised to be unlocked.

Bibliography

[1] Aleksandra Zheleva. *Project: Could You Repeat That, Please? Increasing The Quality And Replicability Of Physiological Virtual Reality Research Through A Comprehensive Experimental Framework*. https://biblio.ugent.be/project/3F015221?limit=5&start=. 2022 (cit. on p. 2).

[2] Jamil Joundi, Klaas Bombeke, Niels Van Kets, Wouter Durnez, Jonas De Bruyne, Glenn Van Wallendael, Peter Lambert, Jelle Saldien, and Lieven De Marez. «ExperienceDNA: A Framework to Conduct and Analyse User Tests in VR Using the Wizard-of-Oz Methodology». In: *Design, User Experience, and Usability: Design for Contemporary Technological Environments: 10th International Conference, DUXU 2021, Held as Part of the 23rd HCI International Conference, HCII 2021, Virtual Event, July 24–29, 2021, Proceedings, Part III*. Berlin, Heidelberg: Springer-Verlag, 2021, pp. 171–186. ISBN: 978-3-030-78226-9. DOI: 10.1007/978-3-030-78227-6_13. URL: https://doi.org/10.1007/978-3-030-78227-6_13 (cit. on pp. 4, 18, 19).

[3] Marcelo Soares, Elizabeth Rosenzweig, and Aaron Marcus. *Design, User Experience, and Usability: Design for Contemporary Technological Environments 10th International Conference, DUXU 2021, Held as Part of the 23rd HCI International Conference, HCII 2021, Virtual Event, July 24–29, 2021, Proceedings, Part III: 10th International Conference, DUXU 2021, Held as Part of the 23rd HCI International Conference, HCII 2021, Virtual Event, July 24–29, 2021, Proceedings, Part III*. Jan. 2021. ISBN: 978-3-030-78226-9. DOI: 10.1007/978-3-030-78227-6 (cit. on p. 4).

[4] Deba Pratim Saha, R. Benjamin Knapp, and Thomas L. Martin. «Affective feedback in a virtual reality based intelligent supermarket». In: *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. UbiComp '17. Maui, Hawaii: Association for Computing Machinery, 2017, pp. 646–653. ISBN: 9781450351904. DOI: 10.1145/3123024.3124426. URL: https://doi.org/10.1145/3123024.3124426 (cit. on p. 6).

[5] Mathias Benedek and Christian Kaernbach. «A continuous measure of phasic electrodermal activity». In: *Journal of Neuroscience Methods* 190.1 (2010), pp. 80–91. ISSN: 0165-0270. DOI: https://doi.org/10.1016/j.jneumeth.2010.04.028. URL: https://www.sciencedirect.com/science/article/pii/S0165027010002335 (cit. on pp. 6, 15).

[6] Raphael Weibel, Jascha Grübel, Hantao Zhao, Tyler Thrash, Dario Meloni, Christoph Hölscher, and Victor Schinazi. «Virtual Reality Experiments with Physiological Measures». In: *Journal of Visualized Experiments* (Aug. 2018), e58318. DOI: 10.3791/58318 (cit. on p. 9).

[7] Elsa Andrea Kirchner, Stephen H. Fairclough, and Frank Kirchner. «Embedded multimodal interfaces in robotics: applications, future trends, and societal implications». In: *The Handbook of Multimodal-Multisensor Interfaces: Language Processing, Software, Commercialization, and Emerging Directions - Volume 3* (2019). URL: https://api.semanticscholar.org/CorpusID:156051062 (cit. on p. 9).

[8] Stephen Fairclough. «A Closed-Loop Perspective on Symbiotic Human-Computer Interaction». In: *Symbiotic Interaction*. Ed. by Benjamin Blankertz, Giulio Jacucci, Luciano Gamberini, Anna Spagnolli, and Jonathan Freeman. Cham: Springer International Publishing, 2015, pp. 57–67. ISBN: 978-3-319-24917-9 (cit. on p. 9).

[9] Stephen Fairclough. «Physiological Computing and Intelligent Adaptation». In: Dec. 2017, pp. 539–556. ISBN: 9780128018514. DOI: 10.1016/B978-0-12-801851-4.00020-3 (cit. on p. 9).

[10] Christopher Baker and Stephen Fairclough. «Adaptive virtual reality». In: Jan. 2022, pp. 159–176. ISBN: 9780128214138. DOI: 10.1016/B978-0-12-821413-8.00014-2 (cit. on p. 10).

[11] Thomas Geijtenbeek, Frans Steenbrink, Egbert Otten, and Oshri Even Zohar. «D-Flow: Immersive virtual reality and real-time feedback for rehabilitation». In: *Proceedings of VRCAI 2011: ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications to Industry* (Dec. 2011). DOI: 10.1145/2087756.2087785 (cit. on pp. 10, 11).

[12] Sean M. Montgomery, Nitin Nair, Phoebe Chen, and Suzanne Dikker. «Introducing EmotiBit, an open-source multi-modal sensor for measuring research-grade physiological signals». In: *Science Talks* 6 (2023), p. 100181. ISSN: 2772-5693. DOI: https://doi.org/10.1016/j.sctalk.2023.100181. URL: https://www.sciencedirect.com/science/article/pii/S2772569323000567 (cit. on p. 13).

[13] Jason J Braithwaite, Diana Patrícia Zethelius Watson, R. O. Jones, and Michael A. Rowe. «Guide for Analysing Electrodermal Activity & Skin Conductance Responses for Psychological Experiments». In: *CTIT technical reports series* (2013). URL: https://api.semanticscholar.org/CorpusID:112717780 (cit. on p. 27).

[14] Siti Farah Hussin, Zunainah Hamid, and Gauri Birasamy. «Design of Butterworth Band-Pass Filter». In: (Jan. 2016), pp. 128–2883 (cit. on p. 34).

[15] David C. Stone. «Application of median filtering to noisy data». In: *Canadian Journal of Chemistry* 73.10 (1995), pp. 1573–1581. DOI: 10.1139/v95-195 (cit. on p. 34).

[16] Zhou Wang and Alan C. Bovik. «Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures». In: *IEEE Signal Processing Magazine* 26.1 (2009), pp. 98–117. DOI: 10.1109/MSP.2008.930649 (cit. on p. 40).

[17] Patrick Schober, Christa Boer, and Lothar Schwarte. «Correlation Coefficients: Appropriate Use and Interpretation». In: *Anesthesia & Analgesia* 126 (Feb. 2018), p. 1. DOI: 10.1213/ANE.0000000000002864 (cit. on p. 43).