Software and Hardware Setup for Emotion Recognition During Video Game Fruition

Marco Granato
Department of Computer Science, University of Milan
Milano, Italy
marco.granato@unimi.it

Davide Gadia
Department of Computer Science, University of Milan
Milano, Italy
gadia@di.unimi.it

Dario Maggiorini
Department of Computer Science, University of Milan
Milano, Italy
dario@di.unimi.it

Laura A. Ripamonti
Department of Computer Science, University of Milan
Milano, Italy
ripamonti@di.unimi.it

ABSTRACT

In Affective Computing research field, different tools and devices are used to acquire physiological data and users' emotional state. Video games, as entertainment software, are developed considering which types of emotions have to be experienced by the players. Unfortunately, the researches on the connection between video games and emotions usually do not provide a quantitative outcome about the emotional impact of different game features. This essay has the main goal to provide an overview on a set of tools that can be used to design an experimental setup aimed to acquire physiological data and players' mental state, using Affective Computing methodology, in video game research. In particular, we propose a hardware architecture to collect physiological data, a software to store, visualize, and synchronize the acquired data (DAPIS), and a tool able to self-assess the users' emotions (ESAT). The overall architecture also involves a method to define game events. The considered physiological data are: electrocardiogram, electromyography, galvanic skin response, and respiration rate. DAPIS provides a real-time visualization of physiological information and a methodology to synchronize the events with ESAT. Lastly, ESAT is a novel method to acquire users' emotional self-assessment in medium-term experiments. After an empirical validation performed on 33 participants, ESAT has proved to be a valid tool which permits to accurately identify the users' emotions.

Goodtechs '18, November 28–30, 2018, Bologna, Italy

ACM ISBN 978-1-4503-6581-9/18/11…$15.00
https://doi.org/10.1145/3284869.3284895

CCS CONCEPTS

• **Human-centered computing** → **Laboratory experiments**; • **Computer systems organization** → **Embedded systems**; • **Software and its engineering** → *Interactive games*;

KEYWORDS

Video Games, Affective Computing, Emotions Recognition, Physiological Data, Players' Emotions

ACM Reference Format:
Marco Granato, Davide Gadia, Dario Maggiorini, and Laura A. Ripamonti. 2018. Software and Hardware Setup for Emotion Recognition During Video Game Fruition. In *International Conference on Smart Objects and Technologies for Social Good (Goodtechs '18), November 28–30, 2018, Bologna, Italy.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3284869.3284895

1 INTRODUCTION

The connection between video games and emotions is a research topic which has received an increasing attention in the last years, since video games are products that aim to provide players' entertainment. Recognizing the players' emotions during video game fruition may help the game designers to propose game features able to maintain the players' engagement. Affective Computing (AC) [10] is a research field which particularly fits with this kind of research, studying, in our specific case, the link between humans emotions and video games. In AC, several datasets have been proposed, each one containing information about users' emotions and a physical correspondence. Usually, these datasets provide the information about participants' emotions during the fruition of a specific set of stimuli. Anyway, at the best of our knowledge, there are not an AC dataset which uses video games as stimuli. In current AC researches, the emotions are usually defined in a continuous space. In particular, 2 or 3 dimensional spaces which involve, respectively, Valence-Arousal (VA), or Pleasure(Valence)-Arousal-Dominance (PAD) [14], are considered. In our research, the main approach used to study players emotions is inspired to the methodologies already presented in other papers on affective databases [2, 12, 13, 15]. However, some characteristic of video game research (e.g., the duration of a game session, the different events that happen in a video game, and the consequent variability in the players' emotional responses) required the design of some novel approaches for the creation of an affective dataset based on the emotional response of video game players.

The application of the tools presented in this paper takes place after the framework validation presented in [7]. Here, the authors have already provided a preliminary overview on the hardware (HW) and the software (SW) developed to store physiological data and to acquire the self-assessment information during video game session. In this paper, we propose architectural improvements and a more accurate explanation of the methodology used to acquire the participants' data. In particular, we have developed a more stable architecture, which also permits the video game fruition through Virtual Reality (VR) headsets. Moreover, we have implemented a new method to avoid the analogical signal noise, and we have corrected some errors that are occurred during the pilot study. Our solution is designed to be applied in affective experiments in video games research, anyway, the framework can be also used for different affective case studies; in particular studies with medium/long experimental sessions (~10/20 minutes) which present different types of events (e.g., a short movie).

The remainder of this paper is organized as follows: in Sec. 2, we provide a brief overview of the devices and the SW already used to collect affective datasets. In Sec. 3, we present the HW architecture used to acquire physiological dataset, and we describe our solution to define the experimental events and the data synchronization; then in Sec. 4 and 5, we present, respectively, the SW designed to store and visualize the physiological information (DAPIS) and the emotion self-assessment tool (ESAT). At the end of the section 5, we also discuss the ESAT validity, starting from a set of experiments. Lastly, in Sec. 6, we provide conclusions and final considerations for future works.

2 RELATED WORKS

In our framework, we need three components: a HW architecture able to reveal human physiological information, a SW able to store the data, and another SW to acquire the emotions self-assessment. In order to acquire a dataset structured by multiple physiological data, the HW has to respect two main requirements: to record in real-time the physiological information, and to synchronize them with a common sample rate. There are some open source and commercial tools able to satisfy these constraints; for sake of brevity, we present only the most common devices. Biosemi[1] and Mindmedia[2] are two companies that sell to the researchers their biofeedback suites. These devices are already successfully used to create physiological databases (e.g., [13, 15]), and they allow to acquire and to synchronize a predefined set of physiological data. For example, NeXus-32, developed by Mindmedia, is designed to reveal electroencephalography (EEG), electromyography (EMG), electrocardiography (ECG), and Electrooculography (EOG). Unfortunately, these devices present limited programmability, and they are often constrained by proprietary communication structure and sensors. Boccignone et al. [2] provides an alternative, using a more suitable methodology. They have used an Arduino Uno, a programmable board, connected to an e-Health Sensor Platform shield[3]. Consequentially, the authors have been able to define a personal communication protocol. However, the Arduino UNO ADC can quantize data at 10-bit precision, a resolution that can be, in particular cases, not sufficient to understand some peculiar physiological features. Furthermore, the e-Health Sensor uses the analogical Arduino pins with presets sensors. As a consequence, the inclusion of additional devices able to acquire physiological data on the board is limited.

Usually, companies that develop devices able to record physiological data also provide a SW to acquire and store the digitized data on a computer. For example, Biosemi has developed *ActiView*, a SW to acquire data collected by *ActiveTwo*[4]. Albeit the SW is an open source project, it is designed to work with the company proprietary device, making the porting on different HW setup a challenge. Others SW (like, e.g., OpenVibe [11]) are designed to support a wide range of devices, anyway, often they are focused on a specific physiological signal. LabView [9] is a general purpose SW which can be used to acquire and elaborate almost all data acquired by sensors. It also supports a wide range of devices, including Arduino boards, and can be programmed by a peculiar visual programming language. However, it is a proprietary SW, with high-level minimum HW requirements.

Lastly, The current approach in emotion tagging applications is to report the users' emotions on *n* vectors (with n equal to 2 or 3). A common implementation is to support the annotators, visualizing in the SW GUI the Self-Assessment Manikins (SAM) [3]. Feeltrace [5], DANTE [2], and ANNEMO [12] are all valid tools used to identify the emotions in Valence and Arousal vectors. Unfortunately, none of these are designed for events management, like, e.g., the beginning and the end of different levels. Moreover, they usually require to tag the emotions on vectors one at a time, a solution

[1]https://www.biosemi.com/
[2]https://www.mindmedia.com/
[3]https://goo.gl/B5qFsy
[4]https://www.biosemi.com/products.htm

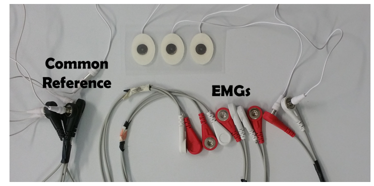
that improves the accuracy but that can be used only for short video sequences.

3 ARCHITECTURE

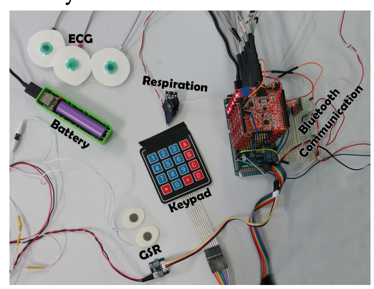
Our research objective is to infer users' emotion using physiological data. Consequentially, the HW architecture has to acquire physiological information, and a set of digital inputs used as "flag values". These values describe different events which happen during the experimental session. Moreover, the device has to provide an analog-to-digital conversion, and to prepare a communication protocol to send data to the computer. Therefore, we have designed a HW architecture based on Arduino Due. It is a programmable board that uses an Atmel SAM3X8E ARM Cortex-M3 CPU with a 32-bit core[5]: its computational effort should guarantee a correct functioning of the data elaboration and communication. The micro-controller has also 12 pins able to read analog information and 54 pins for digital I/O. These pins tolerate a maximum tension of 3.6V, anyway, in our setup, we have used a voltage equal to 3.3V, as suggested in Arduino Due documentation[6]. Moreover, the ADC embedded in the CPU permits a conversion at 12-bit, obtaining, as a consequence, an ADC step equal to 805.66 $\mu V/bit$.

A common problem that affects the analogical signal acquisition is the noise provided by the electric hum of AC current. A solution to avoid this noise is to put a band-stop filter at the electric hum frequency in cascade to the Arduino, or to remove the noise in post processing. However, our solution was to completely isolate the device and sensors from the AC current, feeding the board with an external battery. Moreover, we have avoided the computer current implementing a wireless connection using an Arduino module to convert serial data to Bluetooth (HC-06). It simulates the serial communication using the Bluetooth protocol instead of the cable connection. This solution has provided also a greater flexibility to the overall experimental setup, giving the ability to place the sensors not near to the computer.

Using a set of sensors connected to the Arduino, we have acquired 4 groups of physiological signals: ECG, EMG on 5 facial muscles, Galvanic Skin Response (GSR), Respiration Rate. We have also considered the recent diffusion of VR headsets in gaming applications [6], which lead to a higher level of immersivity in the players' experience. Thus, our architecture is designed to support these devices, recording also the light information produced by lenses of currently available VR headsets. Moreover, as we will describe in the following pages, it is possible to add any analog sensor (e.g., a thermometer to measure the skin temperature) to the architecture. Focusing on physiological signals, we have used



(a) EMG terminal electrodes. The black *snap* of black wires was connected together at a common references, while the others couples of *snaps* (red and white) are placed on the face skin over the muscles that you want to consider.



(b) Overall HW architecture and sensors used to infer physiological data during the experimental sessions.

Figure 1: photo of the HW architecture used to collect the physiological and events data.

6 Olimex EKG-EMG shields[7] to acquire information about ECG and EMG. It is a device already used in biomedical engineering [16], and it is able to read a 3-lead electrode connector via 3.5 jack. Thus, we have connected the leads of an Olimex shield at the wrists and the left ankle, following the guidelines provided by *Einthoven's Triangle* [4], to collect the ECG information using Fiab F9079/100 (36x40 mm) electrodes (Fig. 1b). The other 5 shields have been used to acquire the EMG information up to 5 facial muscles. Lastly, we have connected a common reference at the border of the hair line. We have used small electrodes of dimension 32x32 mm (Fiab F9053N) in order to cover better the face surface, and reduce the vision occlusion. The same electrodes have been used to collect the GSR signal, also named Electrodermal Activity (EDA), connecting them to the middle and ring fingers of the left hand, fingers which are not involved during the experimental session. We have used the Grove GSR sensor, which is equipped also with an amplifier (LM324) in order to improve the information quality provided by the skin potential difference. The last physiological signal is the respiration rate/intensity. It is collected by placing an NTC Thermistor (NTCLE203E3 SB0) [8] under the participant's nose. We also isolate the base of the sensor to avoid the direct contact with

[5]datasheet: https://goo.gl/ZM8zpQ
[6]https://store.arduino.cc/usa/arduino-due

[7]datasheet: https://goo.gl/TDD8UZj
[8]datasheet: http://www.vishay.com/docs/29118/ntcle203.pdf

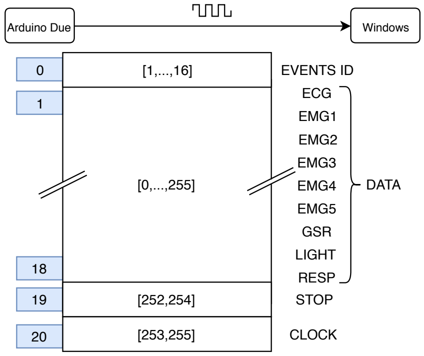


Figure 2: Package structure sent by Arduino to the Computer.

the user's skin, minimizing the noise generated by the epidermis temperature. The thermistor provides an accuracy of ±0.5 C° (between 25 C° and 85 C°), and it reduces the tension when the temperature increases, in our specific case when the user exhales (vice versa when she inhales).

To collect the information on the light presented in VR headset we have used a photoresistor (GL5516) [9].

We have used a 4x4 keypad in order to defines game events. At each key is associated an integer value in a range [0,15], while the value 16 has been used to identify a no event state (i.e., when no buttons have been pressed). In our experiment, the considered events are only the beginning and the end of the game levels, anyway, the architecture is designed to capture information up to 15 events. We have also used two bytes to, respectively, delimit the buffer and clock alert advertise (DEL and CLOCK). The former is used to provide an information of buffer control and to define the end of the data buffer, while the latter is used to define the start of Arduino CPU clock. These bytes are sent after the data quantization performed by the Arduino Due ADC. Furthermore, they modify their values when Arduino starts a new clock, in order to communicate this information to the computer. The delimiter byte is also used to support the clock byte (in case of communication loss), modifying its value according to the CLOCK byte.

The analog data have been converted to digital through the built-in ADC provided by Arduino Due CPU. As we have said, it permits to quantize the data at 12-bit precision. Anyway, the serial communication can send only one byte at a time (2^8). Consequentially, we have split each converted analog data into two bytes, that we have named *h-byte*, and *l-byte*. Thus, we have structured a buffer of 21 bytes, each one transmitted at frequency band of 115200, without parity control and with 1 stop bit (115200/8N1). The package structure is presented in Fig. 2 and it has been transmitted to the

[9]footnote: http://en.nysenba.com/upfiles/file/LDR.pdf

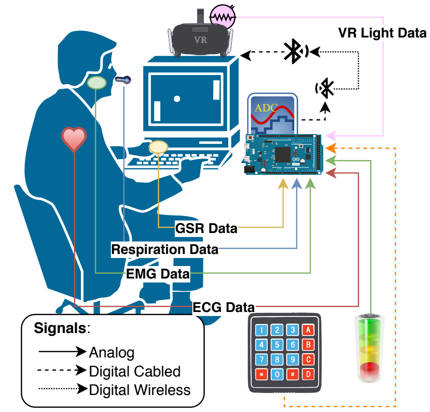


Figure 3: Overview of the HW architecture used to acquire physiological information.

computer at a frequency of 556Hz. The overall architecture is presented in Fig. 3. If a new sensor is added to the HW setup, the acquired information may be appended to the end of DATA area using the guidelines presented in this section, increasing the package size.

4 SOFTWARE FOR PHYSIOLOGICAL DATA ACQUISITION: DAPIS

For our experimental setup, we have developed two different SW: Data Acquisition of Physiological Information Software (DAPIS), a SW developed on the basis of an open-source project[10], used during the experimental phase to acquire, to synchronize, and to visualize the physiological data, and is Emotion Self-Assessment Tool (ESAT), which is used after the video game session to self-assess the participant's emotional states. Both applications are developed in Processing[11], a programming language based on Java, which aims mainly to produce visual contents. For our experimental setup, a video of the screen during the game session is recorded, placing DAPIS GUI in the top-left area of the acquired video. In particular, we have recorded a video at 60 frames per second (fps), which contains the participant's face in the bottom right area, acquired through a camera, and the gameplay, positioned in the top right area (Fig. 5). The DAPIS GUI is structured by 3 main components: the top bar, which presents a set of buttons used to interact with the SW functions, two colored bars used to synchronize the physiological data with the self-assessment information acquired by ESAT, and the central area which visualizes, in real-time, the signal plot. The visualization of the plot can be used to identify, during the experiment, the quality of the acquired signals. DAPIS supports the serial communication,

[10]https://github.com/vsquared/ECG_UNO_Processing3_2_3
[11]https://processing.org/

which is used to exchange packages with external devices. Thus, selecting the correct COM port, the application is able to receive data from the architecture based on Arduino Due presented in Sec. 3. Moreover, the SW automatically writes, in real-time, in a specific path selected by the user, all the acquired physiological data. DAPIS also analyzes the value of the *event flag* (i.e., first byte in the buffer). According to its value, the bars switch their color, permitting the data synchronization with ESAT (Fig. 4). In our specific case, a button is used to identify the beginning of the game level (green bars), while another button defines the level end (red bars). DAPIS contains also different functions able to improve the overall experimental quality. The mains functions are: to change the physiological data visualized in the central area using the keyboard space-bar, to move the center of the plot through the keyboard arrows, to clear the central area, to take a screenshot of the plot, and to record a data baseline of custom duration in a separate file.

5 ESAT: TOOL FOR EMOTION SELF-ASSESSMENT

When the experimental session ends, the video acquired is reproduced into ESAT. In our specific case, the video is structured by three components (Fig. 5): the player face (c), the gameplay (d), and DAPIS GUI (e). ESAT synchronizes its data with the physiological data acquiring the color of the bars presented in the left area of the video. At each color it associates a specific value; in our case study, we have associated 0 for the red bars, and 1 for green. During all the video, the participant interacts with the left and right analog joystick of the same gamepad used to play at the video games to control the self-assessment bars, respectively the left is used to identify valence values (a), while the right is used for arousal (b). To support the self-assessment, we have implemented two tools, SAM [3] and Affective Slider (AS) [1]. A red line (f) underlines the time spent from the beginning of the self-assessment. Moreover, the participants can stop and rollback the video; anyway, during the experiment, we have suggested for each user to limit the uses of these functionalities. We hypothesize that a fluent view of the video can evoke better the emotions experienced during the experimental session. After the SW presentation, we have asked the participant to perform a short training to familiarize with the input system.

Both applications provide in output a *CSV* file containing tables of length $(fps{*}seconds) \times 4$ for ESAT, and $(samplerate{*}seconds) \times 11$ for DAPIS. Moreover, the tools are persistent, thus, they try to minimize the data loss: if a computer crash happens, saving the acquired data at regular intervals. Although functional tests have been performed on a computer with Windows 10 OS, both applications have been developed in a multi-platform language, taking care to produce a SW able to work on the most common operating systems.

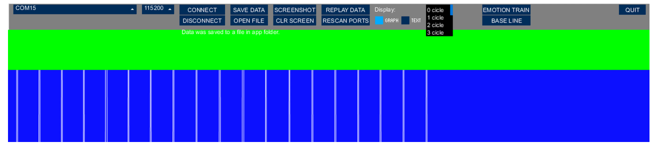


Figure 4: header of DAPIS GUI, it shows the SW functions and the bar color provided by the different events, in particular, the green bar represent the start of game level, while the red bar identify the end.

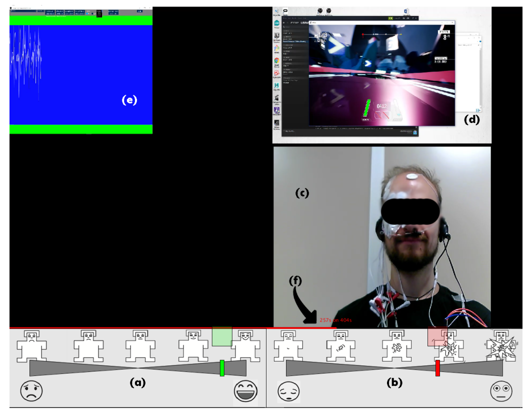


Figure 5: ESAT GUI, used to identify the emotion tagging. It shows the video of a game session and asks to the user to identify hers emotions over all the playback.

ESAT validation

We have also performed a SW validation involving 33 participants (29 males and 4 females), with age between 18 and 40 (μ = 24.66 and σ = 5.15) years old. They have played at two different racing games through a standard monitor and a Virtual Reality (VR) headset (i.e., Oculus Rift DK2). For each participant, a member of laboratory staff has asked to fill out a survey designed to evaluate the overall experiment experience. A subset of the survey questions has been designed to evaluate the average emotion assessment in *valence* and *arousal* vectors, and to evaluate the accurately during the self-assessment stage. The main hypothesis is that the users are able to express their emotional state using ESAT SW, and, as a consequence, that the video tagging procedure can be considered reliable. The participants have declared an average precision in emotion tagging equal to 7.48 (in a rank between 1 to 10) for Arousal, and 7.30 for Valence. Moreover, they have to rank their focus during game session (Arousal) equal 0.56 (in a range between -1 to 1), while the ability of the game to arouse emotions (Valence) is equal to 0.28. In particular, the ability to arouse a positive emotion is ranked equal to 0.34, while 0.14 is the rank in case of a negative emotion.

Considering the data acquired by ESAT, the average values of Arousal and Valence are, respectively, 0.41 ($\sigma = 0.44$), and 0.18 ($\sigma = 0.49$). Considering individual participant outcome, the Mean Square Error (MSE) index between the average value acquired by ESAT and the survey answer is, for Arousal, equal to 0.2655, while for Valence is 0.2858. As a consequence, since the average self-assessed values of arousal and valence are similar to the average survey answers, we can conclude that these data seem to validate ESAT, although the survey answers are slightly overestimated.

6 CONCLUSION

In this paper, we have presented a novel and flexible architecture to acquire human physiological data during video game sessions. Moreover, we have provided a description of two SW developed in order to store the digitized physiological information (DAPIS), and to acquire emotion self-assessment data (EPIS). Lastly, we have empirically demonstrated the validity of EPIS, applying it to a set of physiological data acquired using the proposed HW architecture. The results seem to suggest that EPIS is a valid tool to self-assess the emotions of players during video game fruition, and that the proposed HW setup is an effective solution for the detection of physiological data in the video game research field. The descriptions of the SW and of the HW setup are freely available on GitHub[12].

Starting from the acquired signals, we have extracted a set of features and we have designed an algorithm able to select only the most important subset, as illustrated in [8].

Albeit the tools presented in this paper have been already successfully used in a previous work [7], EPIS can be further improved, for example implementing a dynamic adaptation on the number of physiological data acquired. A similar solution can be also applied on the overall package structure, permitting to place the STOP and CLOCK byte in any position of the stack.

Other improvements can be considered for the ESAT tool, like the support for other input devices (e.g., the signal of two potentiometers connected to an ADC). Lastly, the two SW collect data at two different sample rates: DAPIS samples at the same rate of the Arduino, while ESAT at the video frame rate. Thus, these data, usually, are aligned in post-processing. A future implementation of EPIS can consider a built-in algorithm for signal alignment, in order to interpolate the Arousal/Valence matrix length equal to the physiological data matrix, simplifying the data analysis.

REFERENCES

[1] A. Betella and P.F.M.J. Verschure. 2016. The affective slider: A digital self-assessment scale for the measurement of human emotions. *PLoS ONE* 11, 2 (2016). https://doi.org/10.1371/journal.pone.0148037

[2] Giuseppe Boccignone, Donatello Conte, Vittorio Cuculo, and Raffaella Lanzarotti. 2017. Amhuse: A multimodal dataset for humour sensing. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. ACM, 438–445.

[3] Margaret M Bradley and Peter J Lang. 1994. Measuring emotion: the self-assessment manikin and the semantic differential. *Journal of behavior therapy and experimental psychiatry* 25, 1 (1994), 49–59.

[4] Mary Boudreau Conover. 2003. *Understanding electrocardiography*. Elsevier Health Sciences.

[5] Roddy Cowie, Ellen Douglas-Cowie, Susie Savvidou, Edelle McMahon, Martin Sawey, and Marc Schröder. 2000. 'FEELTRACE': An instrument for recording perceived emotion in real time. In *ISCA tutorial and research workshop (ITRW) on speech and emotion*.

[6] D. Gadia, M. Granato, D. Maggiorini, L.A. Ripamonti, and C. Vismara. 2018. Consumer-oriented Head Mounted Displays: Analysis and Evaluation of Stereoscopic Characteristics and User Preferences. *Mobile Networks and Applications* 23, 1 (2018), 136–146.

[7] M. Granato, D. Gadia, D. Maggiorini, and Ripamonti. 2017. Emotions detection through the analysis of physiological information during video games fruition. *Lecture Notes in Computer Science (Proceedings of 6th International Conference of Games and Learning Alliance - GALA 2017)* 10653, 197–207.

[8] M. Granato, D. Gadia, D. Maggiorini, and Ripamonti. 2018. Feature Extraction and Selection for Real-Time Emotion Recognition in Video Games Players. *Proceedings of the International Workshop on Ubiquitous implicit BIOmetrics and health signals monitoring for person-centric applications (UBIO 2018).*

[9] Cor J Kalkman. 1995. LabVIEW: a software system for data acquisition, data analysis, and instrument control. *Journal of clinical monitoring* 11, 1 (1995), 51–58.

[10] Rosalind W. Picard. 2000. *Affective Computing (The MIT Press)*. The MIT Press.

[11] Yann Renard, Fabien Lotte, Guillaume Gibert, Marco Congedo, Emmanuel Maby, Vincent Delannoy, Olivier Bertrand, and Anatole Lécuyer. 2010. Openvibe: An open-source software platform to design, test, and use brain–computer interfaces in real and virtual environments. *Presence: teleoperators and virtual environments* 19, 1 (2010), 35–53.

[12] Fabien Ringeval, Andreas Sonderegger, Juergen Sauer, and Denis Lalanne. 2013. Introducing the RECOLA multimodal corpus of remote collaborative and affective interactions. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*. IEEE, 1–8.

[13] Stefanie Rukavina, Sascha Gruss, Steffen Walter, Holger Hoffmann, and Harald C Traue. 2015. Open_emorec_ii-a multimodal corpus of human-computer interaction. *International Journal of Computer, Electrical, Automation, Control and Information Engineering* 9, 5 (2015), 977–983.

[14] James A Russell and Albert Mehrabian. 1977. Evidence for a three-factor theory of emotions. *Journal of research in Personality* 11, 3 (1977), 273–294.

[15] Mohammad Soleymani, Jeroen Lichtenauer, Thierry Pun, and Maja Pantic. 2012. A multimodal database for affect recognition and implicit tagging. *IEEE Transactions on Affective Computing* 3, 1 (2012), 42–55.

[16] Radovan Stojanović, Anetta Čaplánová, Živorad Kovačević, Nemanja Nemanja, and Zlatko Bundalo. 2015. Alternative approach to addressing infrastructure needs in biomedical engineering programs (Case of emerging economies). *Folia Medica Facultatis Medicinae Universitatis Saraeviensis* 50, 1 (2015), 29–33.

[12] https://github.com/grano00/VGRDevicesAndTools