The Biocybernetic Loop Engine: An Integrated Tool for Creating Physiologically Adaptive Videogames

J. E. Muñoz , E. R. Gouveia, M. S. Cameirão and S. Bermudez I. Badia
Madeira Interactive Technologies Institute, Funchal, Portugal
Universidade da Madeira, Funchal, Portugal

Keywords: Biocybernetic Loop, Videogames, Adaptation, Software, Visual Scripting, Exergames, Heart Rate, Unity3D.

Abstract: Biocybernetic loops (BLs) are physiological adaptation mechanisms created to augment human-computer interaction by interpreting human behaviour via physiological responses. Because of its inherent complexity, the development of BLs has been mainly utilized within the academic environment, with limited use of physiologically adaptive systems in promising fields such as assistive and gaming technologies. The Biocybernetic Loop Engine (BL Engine) is an integrated software tool designed for an easy creation of physiologically modulated videogames by means of wearable sensors. The BL Engine includes a signal acquisition panel, which facilitates the connectivity of multiple physiological sensors and the processing of their signals, a biocybernetic console to rapidly create and iterate adaptive rules using a visual scripting module, and a game connector tool that ties physiological modulations to game variables. In this paper, we present the BL Engine software architecture, its design and implementation process, as well as a proof-of-concept of the system applied to an exergaming experience aiming to improve cardiorespiratory fitness training in older adults. By developing integrated tools that aid the design and implementation of BLs in videogames, we aim to contribute to the dissemination and widespread use of this approach in the gaming industry and serious gaming applications.

1 INTRODUCTION

Physiological computing systems are designed to capture responses of the central and peripheral nervous systems (Fairclough, 2009). This approach offers a novel input control between users and machines (Fairclough and Gilleade, 2014). One use of such approach is to dynamically adjust systems to challenge or provide assistance to users (Gilleade et al., 2005). The concept of physiologically adaptive systems has been widely developed and documented following the biocybernetic loop (BL) construct. BL utilizes the close-loop control, data analysis, decision making and artificial intelligence from Wiener's cybernetics and applies them to physiological computing (Novikov, 2016). This method has been used for instance to assist pilots by detecting their workload levels (Pope et al., 1995); deliver in an autonomous, timely, consistent and accurate way therapy/drugs to patients (Loeb and Cannesson, 2017) (Mishra and Gazzaley, 2014); adapt difficulty levels in musical learning tasks (Yuksel et al., 2016); and challenge and increase exertion in players during exercising with videogames (exergames) based on real-time cardiac responses (Stach et al., 2009). The use of BL adaptations in videogames has shown that stress, boredom, enjoyment, anxiety, engagement, concentration, and alertness can be effectively used to improve the overall game user experience (Bontchev, 2016). Despite the increasing popularity of BLs among game designers and game user researchers (Pope et al., 2014), its implementation still faces several limitations regarding the integration of physiological sensors, the processing of signals, and the communication between physiological systems and videogames (Novak, 2014).

In this paper, we present the development of the Biocybernetic Loop Engine (BL Engine), a flexible and integrated software tool (from sensing to decision making) to create BLs. Our solution is technology agnostic and can be integrated into any existing software platform. The BL Engine builds on

Muñoz, J., Gouveia, E., Cameirão, M. and Badia, S.
The Biocybernetic Loop Engine: An Integrated Tool for Creating Physiologically Adaptive Videogames.
DOI: 10.5220/0006429800450054
In *Proceedings of the 4th International Conference on Physiological Computing Systems (PhyCS 2017)*, pages 45-54
ISBN: 978-989-758-268-4

top of a solid BL theoretical construct (Serbedzija and Fairclough, 2009) and proposes a more practical and applied adaptation technique. We start by describing the available software platforms that can be used to construct BLs highlighting some of their characteristics, advantages and limitations; then we introduce our BL Engine framework, including software design and implementation processes. Finally, a proof-of-concept experiment is presented using heart rate (HR) based adaptation in a cardiorespiratory fitness exergame.

2 RELATED WORK

The development of BLs has been advanced by academia mainly for research purposes (Pope et al., 2014). Several examples have demonstrated the efficacy of physiological adaptation to improve system automation (Prinzel III et al., 2003), player engagement in gaming experiences (Ewing et al., 2016) and exertion levels in fitness interventions (Ketcheson et al., 2015). Although BLs enable the creation of genuine intelligent systems that use implicit task-context and user-intention information (Jacucci et al., 2015), the creation of such systems is inherently difficult. The fundamental architecture of BLs requires a systematic integration of human-body signals, data conditioning for artefact removal and noise reduction, a feature extraction stage, and a psychophysiological inference process to finally translate data to action (Fairclough and Gilleade, 2012). As a result, this is an arduous process and most of the time the construction of BLs is custom-built for single-task systems, which makes it difficult to replicate or generalize to other applications (Pope et al., 2014). This impedes researchers and developers to rapidly design, construct, iterate and validated new prototypes.

Some software tools have emerged in the last decade to facilitate the creation of BLs and spreading the use of physiological adaptation for multiple purposes. Interestingly, all of them use visual language scripting techniques to simplify the construction process. One well-known example is the OpenViBE software platform (Renard et al., 2010), an open-sourced tool created to support brain-computer interface (BCI) experiments. Using a modular, flexible and simplistic architecture, OpenViBE has been successfully used in closed-loop systems for assistive technology such as spellers, as well as for BCI videogames and virtual reality simulations (Clerc et al., 2016) (Vourvopoulos et al., 2015). Although OpenViBE has been mainly used for BCI applications, studies using ECG data for tangible interfaces (Gervais et al., 2016) showed the potential of the software besides neurophysiological signals. The FlyLoop framework (Peck et al., 2015) is a small and lightweight approach in Java that enables programmers to rapidly develop and experiment with physiologically intelligent systems. The system is presented as a tool to improve decision-making in workload detection via wearable biosensors. Consisting of a set of four modules (data sources, filters, learners, and outputs), the framework is designed to provide reproducibility and accessibility to non-programmer users. Finally, the Neuromore platform was initially designed as a flexible tool to create novel biofeedback visualizations (Jillich, 2014) (Kosch et al., 2016). Nowadays the tool is presented as a development platform for interactive applications which can combine real time physiological data and machine intelligence to create BLs. Focused in the use of wearable and low-cost BCI systems, Neuromore combines multiple technologies to connect commercial-grade physiological sensors with visual scripting. It can process data and classify it in terms of states of mind such as focus, relaxation, flow, creativity or concentration. Unfortunately, the software is still in early stages and the integration with game engines is still unclear.

Some features are still lacking in current software tools to create physiologically adaptive videogames, specifically: a) versatility to support multiple body signals; b) integration with game engines; and c) simplicity to create adaptation rules. To tackle the limitations of the existing technologies we have developed the Biocybernetic Loop Engine.

3 THE BIOCYBERNETIC LOOP ENGINE FRAMEWORK

3.1 Software Design and Development

3.1.1 Design Requirements

The BL engine software was designed to be used by people both with and without specific training in physiological computing and/or programming skills. We identified a list of implementation requirements that guided the design of the BL Engine in the signal acquisition, signal processing and feature extraction, and adaptation domains, as well as its integration with other software systems.

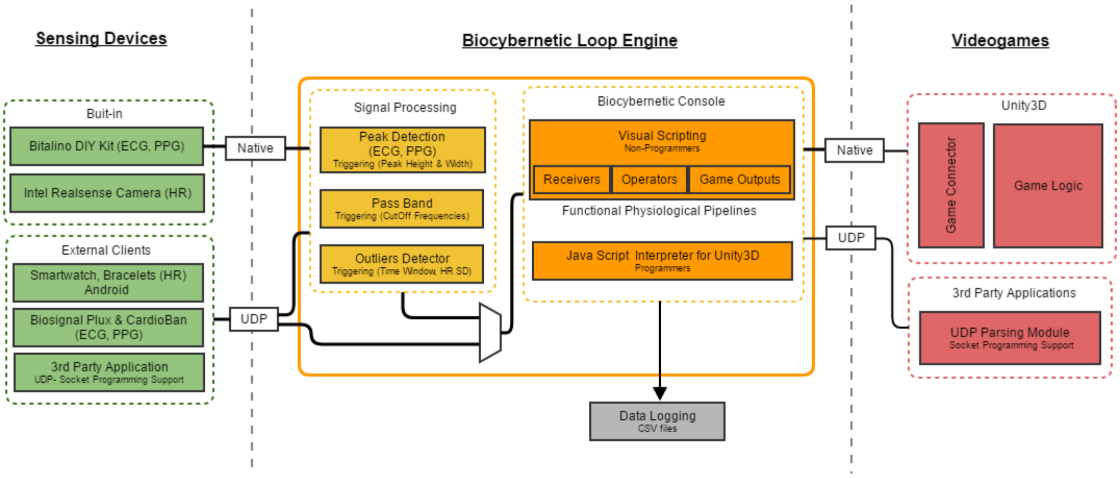

Figure 1: The BL Engine system architecture, which covers the signal acquisition stage via supporting multiple sensing devices and allowing a basic processing of the signals to extract the HR data. It includes a console to create adaptive rules and facilitates the communication with videogames, natively supporting those developed in Unity3D.

Signal Acquisition, Signal Processing and Feature Extraction: one of the biggest limitations when using physiological signals in interactive projects is the connectivity with multiple devices. The lack of standardization of components, different communication protocols and measurements offer a highly variable scenario (Novak, 2014), thus the BL engine should facilitate and streamline the signal acquisition process. Further, the real time signal processing of the acquired signals is an engineering challenge (Jacucci et al., 2015), and thus, the inclusion of common filters to process signals is imperative. Finally, although the features commonly extracted from physiological signals to carry out psychophysiological inferences are relatively well-defined (Cowley et al., 2016), their use for biocybernetic adaptation is still not well understood. Thus, the extraction of meaningful physiological parameters from sensor signals is necessary as they are the main input of the BLs.

Adaptation: the second set of requirements relates to the design of the adaptive rules, which contain the intelligence of the BL system. Essentially, these rules encompass the decision-making process underlying physiological adaptation. Although simple Boolean rules based on if/then rules have been successfully used in past investigations (Karran et al., 2015), more advanced techniques based on proportional-integral-derivative control (Parnandi et al., 2013) and machine learning approaches (Verhulst et al., 2015) have also shown encouraging results. Despite those advances, the implementation and iteration of adaptive rules in BLs require extensive reprogramming processes in order to create playable prototypes (Pope et al., 2014). Consequently, our BL engine should embrace an agile methodology that facilitates the generation of adaptive rules and enables a fast iteration on them.

Integration: finally, a full integration with third party software systems, such as videogames, is required. Even though excellent game engines are freely available (e.g. Unity3D, Unreal Engine), the integration of physiological computing technologies in those systems is not a simple task due to the lack of standardized and functional signal processing toolboxes (Bontchev, 2016). Only few examples enable the integration of physiological sensors with the Unity3D game engine such as the PhysSigTK (Rank and Lu, 2015), RehabNet CP (Vourvopoulos et al., 2013) and PhysioVR framework (Muñoz et al., 2016). However, BLs require not only a simple integration of sensors but also a bi-directional communication between the extracted physiological parameters and the videogame variables in real time.

3.1.2 Design Process

The BL Engine aims to be an extensive tool for the creation of BLs in multiple dimensions such as cardiac, muscular, emotional or motor domains. At this stage we developed the cardiac module and tested its functionality implementing a BL in gaming applications. We used multiple techniques from software engineering for the development - process workflow understanding, activities and system dynamics visualization relying on flow and UML diagrams of the system, and low fidelity prototyping through digital interactive wireframes-prior to its implementation.

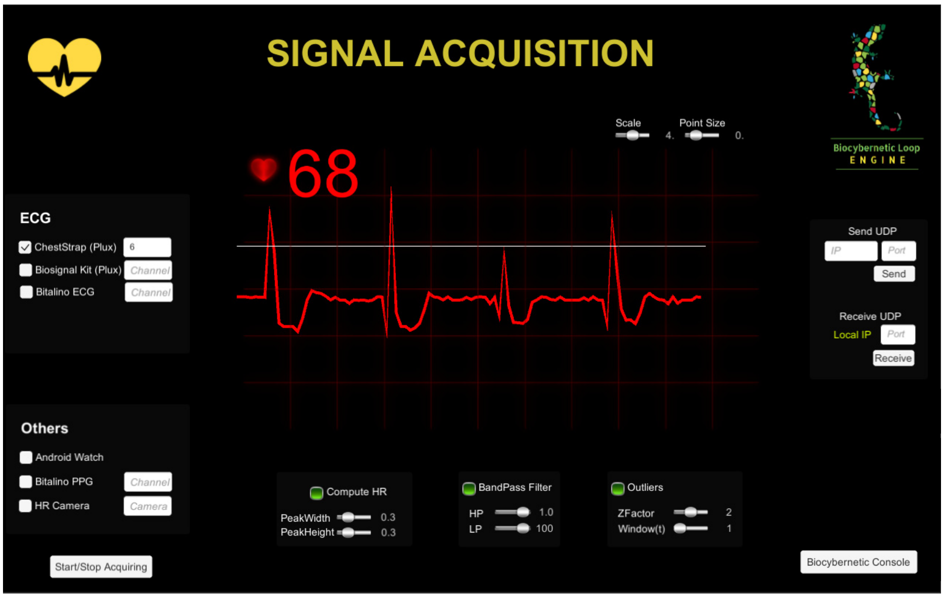

Figure 2: Screenshot of the Signal Acquisition Panel in the BL Engine. The image shows an ECG signal from the CardioBan chest strap (PLUX, Lisbon, Portugal) with the computed HR. Options for band-pass filtering and outliers' detection are activated to improve the HR computation.

3.1.3 Implementation

The BL Engine is a software tool implemented in Unity 3D (Unity Technologies, San Francisco, USA) and it is composed by 3 main modules: a) the signal acquisition panel, b) the biocybernetic console, and c) the game connector. Using the BL Engine, users are able to easily design physiological adaptations of their videogames following the complete processing pipeline from physiological data collection, analysis to the final translation in videogames (see figure 1).

Signal Acquisition Panel: the BL Engine supports the acquisition of a basic range of wearable devices (figure 2) including the Biosignal Plux (PLUX, Lisbon, Portugal), a professional biosignal acquisition kit with 8-12 bit resolution and 1000Hz sampling rate that measures blood volume pressure through photopletysmography (PPG), electrocardiography (ECG), electromyography (EMG), electrodermal activity (EDA), and respiration; a chest strap sensor called CardioBan (PLUX, Lisbon, Portugal) with integrated ECG, respiration and acceleration sensors; Bitalino (PLUX, Lisbon, Portugal), a low cost DIY biosignal board with ECG, EMG, light intensity, and acceleration sensing; HR data streamed from the RealSense Intel camera (Intel, California, USA); and HR data through Android Wear devices such as smartwatches and wristbands through the PhysioVR App (Muñoz et al., 2016). UDP is used for the communication of the Bioplux, CardioBan and the android wearables through external applications streaming it in the RehabNet protocol (Vourvopoulos et al., 2013), while the Bitalino integration is natively supported by the BL Engine through a serial port interface over a bluetooth connection.

The signal acquisition panel also includes a signal visualization to facilitate the real time data analysis and the feature extraction from data. Two cardiac-related signals can be processed at this stage: PPG and ECG. The acquisition panel includes algorithms for the HR computation based on an adaptive peak-detection technique. Both the peak width and the peak height can be manually adjusted, and adjustable band-pass filters can be used to improve the accuracy of the HR computation. Finally, after the HR computation, an outlier detection algorithm is used based on the following statistical descriptors:

$$Outliers = abs(X - \bar{X}\,) > (zFactor * \sigma(X)\,) \tag{1}$$

where X are the HR measurements in a temporal window, $\bar{X}$ is the mean value of X, and $\sigma(X)$ the standard deviation of X. *zFactor* is a constant with a default value of 2, meaning that every HR data point that differs by more than 2 standard deviations from the previous data point will be considered an outlier. The filtered HR data is then sent to the Biocybernetic Console for further processing.

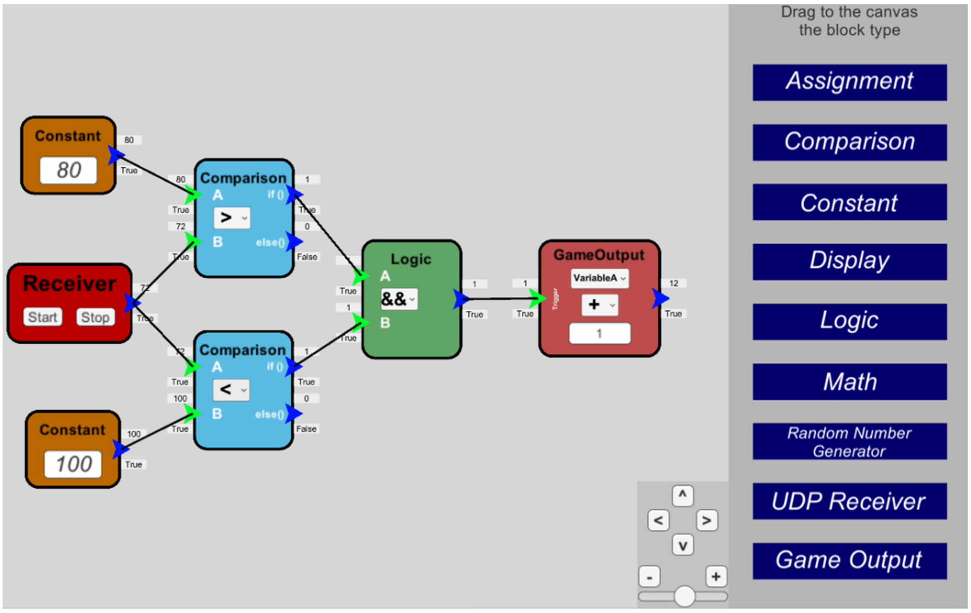

Figure 3: A screenshot of the Biocybernetic Console in the BL Engine representing an adaptation rule in which the game *Variable A* is increased by a value of one (1) once the HR is between 80 and 100 BPMs.

Biocybernetic Console: this console is designed to facilitate the construction of the adaptive rules that use the output computed by the signal acquisition panel to influence the videogame functioning (figure 3). This is achieved through a visual scripting module, which comprises the use of pre-programmed boxes that can be graphically connected to create adaptive rules through functional physiological pipelines (FPP). The blocks can be dragged-and-dropped from the right-side canvas to the left workspace, and inputs and outputs of the boxes can be connected drawing connecting lines. Additionally, the workspace size can be modified allowing the creation of multiple FPPs that can run in parallel using inputs from different physiological sensors. The blocks fall into three different categories:

- Receivers: blocks that receive and/or simulate data. Here, we can even receive data directly from the signal acquisition panel or data coming from any external application supporting socket programming through the UDP Reh@Net protocol.
- Operators: blocks that make comparisons, mathematical and logical operations, and variables' assignments. Blocks for adding constants and visualizing results are also implemented.
- Game outputs: blocks for modifying game variables in real time. Game variables are exposed to the BL Engine using the Unity3D Game Connector module or through UDP for third-party applications.

Finally, to expand the possibilities of creating adaptation rules, the biocybernetic console includes a JavaScript interpreter that allows the generation of more complex adaptation rules directly coding them and executing them on-the-fly. After the rule creation process, users can test its behaviour in real-time and iterate with multiple adaptive rules during run-time. Data from both the BL Engine and the videogame can be synchronously recorded for post-processing using a CSV data writer script.

Game Connector: to enable the connectivity between the BL Engine and the videogames, we provide the game connector module, which is wrapped into a prefabricated package (prefab) that can be integrated in any videogame developed in Unity 3D. The Unity prefab package contains the scripts needed for bidirectional communication with the BL Engine. The connector receives the physiological data via UDP communication, makes specific videogame variables available to the Biocybernetic Console for the creation of the adaptation rules, and updates them in real time accordingly. Any third party application supporting socket programming (such as Unreal Engine and others) can also receive data from the biocybernetic console via a UDP parsing module.

4 PRELIMINARY ASSESSMENT

With the objective of assessing the feasibility of using the BL Engine technology to create physiologically adaptive videogames, we presented a case study using HR data in an Exergaming experience.

4.1 Physiologically Adaptive Exerpong

4.1.1 Exergame Design

Exerpong is an exergame developed in Unity3D and designed for agility and balance training in active seniors (Muñoz J.E. et al., 2016). The exergame was designed as an adaptation of the classic 2D Pong in which the goal is to hit a ball using a virtual paddle. We used the BL Engine to make adaptations based on the HR of users during Exerpong gameplay, with the goal of driving their HR to a target zone. The target HR zone is expressed in terms of the heart rate reserve (HRR) which is the difference between the maximum HR and the HR during the resting state. In this zone, the health benefits of a cardiorespiratory training session can be maximized via stressing the cardiac muscle without over-exercised it (Heyward and Gibson, 2014). For older adults, the ACSM recommends exercise at 40% to 70% of the HRR for moderate intensities (Rahl, 2010) during sessions of 20 to 30 minutes.

Game parameters are adjusted following the dual flow model for exergaming (Sinclair et al., 2009), in our case to adapt for *Gameplay* and *CardioRespiratory Fitness*. *Gameplay* adaptation changes games parameters to improve game attractiveness and balance the challenge in the following way:

- The paddle size decreases once the player hits the ball and increases once he/she misses it.
- The ball velocity decreases if the player misses three consecutive balls.

The *CardioRespiratory Fitness* component of the Exerpong is adapted using the real time HR data according to the following rule:

- The ball velocity increases if the 30-seconds HR average is under 50% of the HRR, and decreases otherwise.

4.1.2 Experimental Setup

A white 2.5m x 3.0m PVC surface was used to project Exerpong on the floor. The KinectV2 sensor (Microsoft, Washington, USA) was used for tracking the user's waist position and mapping it to the paddle position. A smartwatch Motorola 360 Sport was used to capture HR data at 1 Hz sampling frequency as input for the physiological adaptation. ECG signals were recorded using the Biosignal Plux at 1000 Hz through a triode dry electrode placed on the V_2 pre-cordial derivation. An extra-elastic band was used to reinforce the contact and stability between the electrodes and the skin during the exercise. The figure 4 shows the experimental setup.

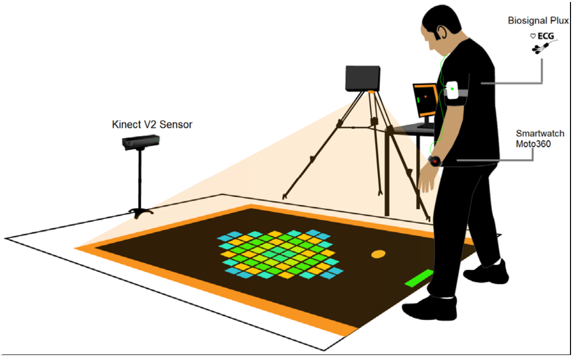

Figure 4: Diagram illustrating the experiment setup for the adaptive Exerpong consisting of a KinectV2 sensor, a projected environment and the physiological sensors.

The HR computed via the post-processing of ECG signals was used as a ground-truth and compared with the smartwatch data.

4.2 Case Study

A community-dwelling 62-years-old female participated in the study. The participant was recruited at a local senior sports facility. The Montreal Cognitive Assessment (MoCA) (Freitas et al., 2011) was used for cognitive screening. The participant scored 29/30, indicating normal cognitive function to understand the instructions and participate in the experiment. The level of physical activity was assessed using the short version of the International Physical Activity Questionnaire (IPAQ) (Booth et al., 2003), through which the user was categorized for moderate intensity of physical activity. The body mass index was 24 kg/m^2, indicating normal weight. The HR during a 5 minutes resting period was calculated as 71 BPMs, while the HR maximum was estimated to be 164 BPMs following Tanaka's formula (Tanaka et al., 2001). Then, the target HR value was established as 117 BPMs (50% of HRR).

4.3 Protocol

After arrival, the participant received the information about the study, signed an informed consent, provided the demographic information, and underwent IPAQ and MoCA assessments. Subsequently, the participant was asked to remain 5 minutes seated for collection of HR data during resting. ECG signals and the HR from the smartwatch were collected synchronously. A short stretching routine was used to facilitate the muscle and tendons exertion of the lower limbs. The game mechanics of the Exerpong were explained before starting. The value for ball velocity started from the minimum and changed every 30 seconds following the BL Engine adaptations described before. Initial parameters of the user such as age, HR during resting and target HR percentage were configured for the adaptation. The interaction with the adaptive Exerpong lasted 20 minutes.

4.4 Results

In order to validate the use of the smartwatch as real-time input to the BL Engine, the post-processed HR data from the ECG signals was used for comparison with the data from the smartwatch. A low root-mean-square-error of 5 BPMs was computed, which is in accordance with previously reported values (Mike Prospero, 2016).

Figure 5 shows the resulting *Gameplay* and *CardioRespiratory Fitness* adaptations during exercising with the physiologically adaptive Exerpong. Figure 5A shows the HR measurements of the user during the complete session, and the red line indicates the target HR value (117 BPMs). We can observe from the data that after approximately six minutes of training the user reached the expected value. Figure 5B shows that this happens as a result of a constant increase in ball velocity. Through playing ExerPong, the user achieved an average HR of 116 BPMs, very close to the intended target HR. This value is considerably higher than the registered average HR in a conventional training session in the same senior gym, which is 93 BPMs.

Figure 5B shows the dynamics of the *CardioRespiratory Fitness* adaptations by the BL Engine to modify the ball velocity. The result is a classic bang-bang controller switching between +1 and -1 values depending on the HR value relative to the target. It can be observed that despite the binary decisions, the BL Engine successfully engaged the user and modulated her HR oscillating around the target value, crossing it multiple times (minutes 6, 10, 12, 14, 18) during the session as reaction to the game adaptations. Finally, Figure 5C shows the behaviour of the two game variables that were

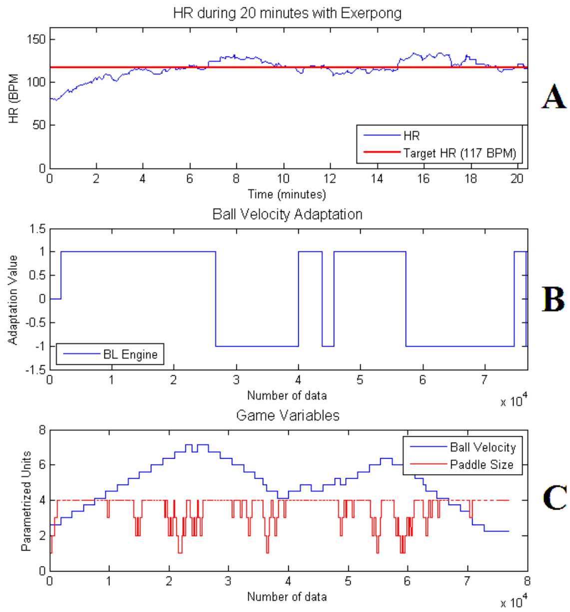

Figure 5: Adaptation during the session with Exerpong. **A**: HR responses (blue line) and target heart rate (red line); **B**: adaptation values (+1 or -1) sent from the BL Engine to increase or decrease the ball velocity; **C**: behavior of both paddle size and ball velocity game parameters.

modified by both *Gameplay* and *CardioRespiratory Fitness* adaptations. It can be observed that the ball speed (blue line) acted as the main driver for the changes in HR values. A cross-correlation analysis of those two variables showed a very high similarity (0.82). Moreover, the paddle size (red line) was being reduced by the *Gameplay* adaptations, making the task more challenging and encouraging movement performance, thus facilitating the modulation of HR responses.

5 DISCUSSION & CONCLUSION

This paper presented the design, implementation and a proof-of-concept of the Biocybernetic Loop Engine tool, which is freely available at: http://neurorehabilitation.m-iti.org/tools/blengine.
The here presented proof-of-concept experiment evaluated the feasibility of including BLs to adapt exergaming experiences for the maximization of its effectiveness for cardiorespiratory training in seniors. This is maybe one of the more complex scenarios for real-time HR-based adaptation, since monitoring cardiac responses during exercise is particularly challenging due to movement artefacts.

The use of tools such as the BL Engine will facilitate a better understanding of the role of BLs in gaming technologies, a more streamlined connectivity with physiological sensors, a fast iteration of adaptation techniques, and an easy integration of physiological intelligence in videogames. The BL Engine addresses such challenges through relatively low-cost and wearable physiological sensors such as smartwatches, utilizes a fully functional and modular user interface, integrates a visual scripting module, which facilitates programming the adaptation rules, and provides tools for a simplistic integration of any videogame developed in Unity3D. Furthermore, the system provides a integrated and comprehensible architecture which might facilitate the incorporation of multiple physiological features streamed from several sensors and captured in the biocybernetic console, hence permitting the conception of multimodal BLs (D'mello and Kory, 2015).

Novel physiologically modulated videogames might overcome the existing limitations and become part of our daily activities through systems such as exergames for exercise prescription or interactive applications for stress management. This will bring uncountable benefits in augmenting human computer interactions. Through the integration of physiological adaptation, more affective and personalized videogames can be developed enabling a fluent communication between the physiological parameters and the videogame variables.

6 FUTURE WORK

Currently, we are planning a cross-sectional study with the adaptive Exerpong in a group of senior users for evaluating the appropriateness of the adaptation for boosting effectiveness in exergaming-based interventions for exercise promotion. In addition, a longitudinal intervention will be conducted, to compare the effectiveness of such approach in comparison to traditional physical exercise activities.

An interesting future application of the BL Engine could be the creation of adaptive rules based on heart rate variability (HRV) analysis using specific measurements which have been associated to workload and stress. For instance, novel serious videogames for stress management can react dynamically to HRV parameters such as the SDNN (standard deviation of normal R-R intervals). This will provide a very compelling scenario for train the awareness of one's internal physiological states (also called interoceptive awareness) (Schulz and Vögele, 2015), which might be one of the cornerstones in the wellbeing upsurge via physiological computing technologies (Critchley et al., 2004). Finally, we believe that the BLs have high potential for being integrated with virtual reality applications, opening up a new communication pathway for adaptive contents creation (Siriborvornratanakul, 2016).

Although the BL Engine only contains the cardiorespiratory module at this stage, the simplified data collection and analysis and translation model can be transversally used for physiological adaptations besides cardiac-related signals.

7 CONTRIBUTIONS

JEM and SBB defined and designed the BL Engine. JEM implemented the software. All authors defined the experimental protocol. JEM collected and analyzed the data. All authors interpreted the results. All authors revised and approved the current version of the manuscript.

ACKNOWLEDGEMENTS

The authors would like to thank Teresa Paulino for developing the Exerpong, and for contributing to the development of the signal acquisition panel, the game connector, and the final integration of the system; and Luis Quintero for contributing to the development of the visual scripting module for the biocybernetic console. This work was supported by the Portuguese Foundation for Science and Technology through the Augmented Human Assistance project (CMUP-ERI/HCI/0046/2013), Projeto Estratégico UID/EEA/50009/2013, and ARDITI (Agência Regional para o Desenvolvimento da Investigação, Tecnologia e Inovação).

REFERENCES

Bontchev, B., 2016. Adaptation in Affective Video Games: A Literature Review. C*ybern. Inf. Technol. 16, 3–34.*

Booth, M. L., Ainsworth, B. E., Pratt, M., Ekelund, U., Yngve, A., Sallis, J. F., Oja, P., 2003. *International physical activity questionnaire: 12-country reliability and validity. Med Sci Sports Exerc 195*, 3508–1381.

Clerc, M., Bougrain, L., Lotte, F., 2016. Brain-Computer Interfaces 2: Technology and Applications. John Wiley & Sons.

Cowley, B., Filetti, M., Lukander, K., Torniainen, J., Henelius, A., Ahonen, L., Barral, O., Kosunen, I., Valtonen, T., Huotilainen, M., others, 2016. The Psychophysiology Primer: A Guide to Methods and a Broad Review with a Focus on Human–Computer Interaction. Found. *Trends® Human–Computer Interact. 9, 151–308.*

Critchley, H. D., Wiens, S., Rotshtein, P., Öhman, A., Dolan, R. J., 2004. Neural systems supporting interoceptive awareness. Nat. Neurosci. 7, 189–195.

D'Mello, S. K., Kory, J., 2015. A review and meta-analysis of multimodal affect detection systems. ACM Comput. Surv. CSUR 47, 43.

Ewing, K. C., Fairclough, S. H., Gilleade, K., 2016. Evaluation of an Adaptive Game that Uses EEG Measures Validated during the Design Process as Inputs to a Biocybernetic Loop. Front. Hum. Neurosci. 10.

Fairclough, S., Gilleade, K., 2012. Construction of the Biocybernetic Loop: A Case Study, in: Proceedings of *the 14th ACM International Conference on Multimodal Interaction, ICMI '12. ACM, New York, NY, USA,* pp. 571–578. doi: 10.1145/2388676. 2388797.

Fairclough, S. H., 2009. Fundamentals of physiological computing. Interact. Comput. 21, 133–145.

Fairclough, S. H., Gilleade, K., 2014. Advances in physiological computing. *Springer*.

Freitas, S., Simões, M.R., Alves, L., Santana, I., 2011. Montreal Cognitive Assessment (MoCA): normative study for the Portuguese population. J. Clin. Exp. Neuropsychol. 33, 989–996.

Gervais, R., Frey, J., Gay, A., Lotte, F., Hachet, M., 2016. Tobe: Tangible out-of-body experience, in: *Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction. ACM,* pp. 227–235.

Gilleade, K., Dix, A., Allanson, J., 2005. Affective Videogames and Modes of Affective Gaming: Assist Me, Challenge Me, Emote Me. Presented at the *DiGRA 2005: Changing Views – Worlds in Play.*

Heyward, V. H., Gibson, A., 2014. Advanced Fitness Assessment and Exercise Prescription 7th Edition. Human Kinetics.

Jacucci, G., Fairclough, S., Solovey, E.T., 2015. Physiological Computing. Computer 48, 12–16. doi:10.1109/MC.2015.291.

Jillich, B., 2014. Acquisition, analysis and visualization of data from physiological sensors for biofeedback applications.

Karran, A. J., Fairclough, S. H., Gilleade, K., 2015. A framework for psychophysiological classification within a cultural heritage context using interest. *ACM Trans. Comput.-Hum. Interact. TOCHI 21, 34.*

Ketcheson, M., Ye, Z., Graham, T.C.N., 2015. Designing for Exertion: How Heart-Rate Power-ups Increase Physical Activity in Exergames, in: *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play, CHI PLAY '15. ACM, New York, NY, USA*, pp. 79–89. doi:10.1145/2793107.2793122.

Kosch, T., Hassib, M., Schmidt, A., 2016. The Brain Matters: A 3D Real-Time Visualization to Examine Brain Source Activation Leveraging Neurofeedback, in: *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. ACM*, pp. 1570–1576.

Loeb, R. G., Cannesson, M., 2017. Closed-Loop Anesthesia: Ready for Prime Time? LWW.

Mike Prospero, 2016. Who Has The Most Accurate Heart Rate Monitor? *Tomsguide*.

Mishra, J., Gazzaley, A., 2014. Closed-loop rehabilitation of age-related cognitive disorders, in: Seminars in Neurology. *Thieme Medical Publishers,* pp. 584–590.

Muñoz J. E., Bermudez S., Rubio E., Cameirao M., 2016. Modulation of Physiological Responses and Activity Levels During Exergame Experiences, in: *2016 18th International Conference on Virtual Worlds and Games for Serious Applications. IEEE*, p. In press.

Muñoz, J. E., Paulino, T., Vasanth, H., Baras, K., 2016. PhysioVR: A novel mobile virtual reality framework for physiological computing, in: *E-Health Networking, Applications and Services (Healthcom), 2016 IEEE 18th International Conference on. IEEE*, pp. 1–6.

Novak, D., 2014. Engineering Issues in Physiological Computing, in: Advances in Physiological Computing. Springer, pp. 17–38.

Novikov, D. A., 2016. Cybernetics in the 20th Century, in: Cybernetics. *Springer*, pp. 1–19.

Parnandi, A., Son, Y., Gutierrez-Osuna, R., 2013. A Control-Theoretic Approach to Adaptive Physiological Games, in: 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction. Presented at the *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction,* pp. 7–12. doi: 10.1109/ACII. 2013.8.

Peck, E. M., Easse, E., Marshall, N., Stratton, W., Perrone, L. F., 2015. FlyLoop: a micro framework for rapid development of physiological computing systems, in: *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems. ACM,* pp. 152–157.

Pope, A. T., Bogart, E. H., Bartolome, D. S., 1995. Biocybernetic system evaluates indices of operator engagement in automated task. Biol. Psychol. 40, 187–195.

Pope, A. T., Stephens, C. L., Gilleade, K., 2014. Biocybernetic Adaptation as Biofeedback Training Method, in: Fairclough, S. H., Gilleade, K. (Eds.), *Advances in Physiological Computing, Human–Computer Interaction Series. Springer London,* pp. 91–115.

Prinzel III, L. J., Parasuraman, R., Freeman, F. G., Scerbo, M. W., Mikulka, P.J., Pope, A.T., 2003. Three experiments examining the use of electroencephalogram, event-related potentials, and heart-rate variability for real-time human-centered adaptive automation design.

Rahl, R. L., 2010. Physical activity and health guidelines. Recomm. Var. Ages Fit. Levels Cond. From 57.

Rank, S., Lu, C., 2015. PhysSigTK: Enabling engagement experiments with physiological signals for game design, in: *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on. IEEE*, pp. 968–969.

Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., Bertrand, O., Lécuyer, A., 2010. OpenViBE: an open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments. Presence Teleoperators Virtual Environ. 19, 35–53.

Schulz, A., Vögele, C., 2015. Interoception and stress. Front. Psychol. 6, 993.

Serbedzija, N. B., Fairclough, S.H., 2009. Biocybernetic loop: From awareness to evolution, in: *Evolutionary Computation, 2009. CEC'09. IEEE Congress on. IEEE,* pp. 2063–2069.

Sinclair, J., Hingston, P., Masek, M., 2009. Exergame development using the dual flow model, in: *Proceedings of the Sixth Australasian Conference on Interactive Entertainment. ACM*, p. 11.

Siriborvornratanakul, T., 2016. A Study of Virtual Reality Headsets and Physiological Extension Possibilities, in: *International Conference on Computational Science and Its Applications. Springer,* pp. 497–508.

Stach, T., Graham, T. C., Yim, J., Rhodes, R.E., 2009. Heart rate control of exercise video games, in: Proceedings of Graphics Interface 2009. *Canadian Information Processing Society,* pp. 125–132.

Tanaka, H., Monahan, K. D., Seals, D.R., 2001. Age-predicted maximal heart rate revisited. J. Am. Coll. Cardiol. 37, 153–156.

Verhulst, A., Yamaguchi, T., Richard, P., 2015. Physiological-based Dynamic Difficulty Adaptation in a Theragame for Children with Cerebral Palsy., in: PhyCS. pp. 164–171.

Vourvopoulos, A., Cardona, J.E.M., Bermudez i Badia, S., 2015. Optimizing motor imagery neurofeedback through the use of multimodal immersive virtual reality and motor priming, in: *Virtual Rehabilitation Proceedings (ICVR), 2015 International Conference on. IEEE*, pp. 228–234.

Vourvopoulos, A., Faria, A. L., Cameirão, M.S., Bermudez i Badia, S., 2013. RehabNet: A distributed architecture for motor and cognitive neuro-rehabilitation, in: 2013 *IEEE 15th International Conference on E-Health Networking, Applications Services (Healthcom)*. pp. 454–459.

Yuksel, B. F., Oleson, K.B., Harrison, L., Peck, E.M., Afergan, D., Chang, R., Jacob, R.J., 2016. Learn piano with BACh: An adaptive learning interface that adjusts task difficulty based on brain state, in: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. ACM,* pp. 5372–5384.