A Communication Middleware for Developing Serious Games that use Personal Health Devices

Douglas Battisti, Luma Wanderley de Oliveira, Marcos Alves Vieira, Sérgio T Carvalho
Instituto de Informática (INF)
Universidade Federal de Goiás (UFG)
Goiânia, Goiás, Brasil
Email: dougbattisti.5@gmail.com, luma.lwo@gmail.com, marconian@gmail.com, sergio@inf.ufg.br

***Abstract*—In addition to their entertainment value, Serious Games can have beneficial therapeutic effects for older people, improving their health and well-being. Like Serious Games, continuous monitoring of the elderly's physiological data through Personal Health Devices (PHD) (e.g., blood pressure device) has become an ally to understand patient involvement with treatment. Solutions that combine Serious Games along with this type of device promotes a personalized experience in therapeutic terms. However, the development of games that use PHDs is a rather complex task, since it is up to the developer to understand and implement the communication protocol between the game engine and the PHDs. This ongoing work aims to build a Middleware to facilitate communication with the PHDs by adopting the ISO/IEEE 11073, a widely used standard communication. Our proposal also includes an API so developers can reuse the solution. Aiming at a validation of the Middleware, we create a plugin in Unreal Engine 4 capable of communicating with the our Middleware.**

***Keywords*-Serious Games, Personal Health Devices, Game Engines, Middleware, ISO/IEEE 11073 PHD, Unreal Engine 4**

I. Introduction

The beneficial role of Serious Games (SGs) for older people is gradually being accepted as a new tool for treatments and therapies [1] [2]. When SGs promote physical activity and exercise (exergames), in addition to improving physical fitness, they also have the potential to positively change mood, improve confidence with daily functional activities, and improve overall quality of life [3] [4].

Like SGs, continuous monitoring of the patient's behavior and physiological data has become a major ally for home treatment and therapies [5]. This has only become feasible due to the avalanche of Personal Health Devices (PHD) and low-cost sensors that support connectivity and data sharing. Non-intrusion is the key to its wide acceptance by the elderly, especially when it comes to installation in the home environment and use throughout the day [6].

The use of SGs in the home environment has its motivation, but besides the beneficial power of the game, it is necessary to monitor the patient to understand their health conditions. Such information that will customize the game and also the treatment of the patient at home. In this scenario, the physiological data can both customize the experience during the game session and also inform the patient's involvement with the treatment. However, in order for the game to receive and transmit PHD data, a communication solution between the game and the PHDs is required.

When a SG developer is using a game engine, such as Unreal Engine 4 (UE4)[1] or Unity Engine[2], and wishes to use PHDs, comes across with two very distinct elements: the PHD to capture the physiological data of the player and the game engine to aid in the construction of the game. This relationship is illustrated in Figure 1. The game developer needs the connection between these two elements, i.e., the game engine must communicate with the PHDs and the PHDs must communicate with the game engine.

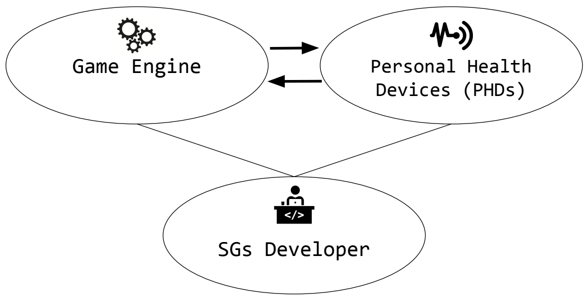

Figure 1. Relationship between the SGs Developer, Game Engine and PHDs.

In general, the developer must adapt the game engine to request and receive the physiological data from the PHDs. This adaptation must be done with the implementation of new mechanisms in the game engine, so that they can communicate with the PHDs. This task involves learning the communication protocol used by the PHDs and the adequacy of the operation of a game engine.

The game itself, with its logic and rules, is a laborious product. For the developer, having a solution that implements the fundamental, such as graphical rendering system and collision system, makes the work more effective. In this

[1] http://www.unrealengine.com
[2] http://www.unity3d.com

sense, this paper aims to propose a middleware in order to facilitate communication with the PHDs, so that the developer can have at his disposal the fundamental in this type of communication. The proposal includes the construction of an API so developers can reuse the solution, and the adoption of the ISO/IEEE 11073 standard for PHDs (X73-PHD)[3] to make the proposed architecture interoperable with the PHDs, since a large number of them is available.

The remainder of this paper is composed of five sections: Section II discusses related works; Section III presents the standard X73-PHD, adopted in the solution; Section IV presents the Middleware architecture; Section V presents the results obtained; and finally Section VI draws conclusions and future work.

II. Related Work

A bibliographic review of papers related to the topic was carried out. There are papers proposing middlewares who implement the X73-PHD standard, others evaluate the use of exergames for the elderly, others discuss the use of physiological data of the player to customize the game session and some discuss the use of the data captured during session to clinically assess the player. The main works are presented as follows.

In [7] a middleware for interoperable PHD communication with Android[4] applications is proposed. The middleware's operation mode shows the necessary points to facilitate access to the collected data. This middleware is concerned with organizing, selecting and communicating with each device, and it is only up to the developer to request the physiological data that wishes.

In [8] the possibilities of using Internet of Things (IoT) accompanied by SGs in the context of personalized health are discussed. This paper proposes a two-way path between SGs and IoT, that is, SGs consuming information from IoT devices to customize the player experience and IoT devices consuming information from SGs for personalization of the environment. The performance during the game can be justified with physiological data. Important decisions can be made before starting the game, such as decreasing the intensity of the exercise due to high heart pressure in the last hours.

In [3] an exergaming platform called FitForAll (FFA) is presented, which offers physical exercises specifically for the elderly. Physical exercises are presented in an immersive game environment, aiming to promote adherence to the physical exercise protocol. Figure 2 illustrates some of the exercises built into the platform. The work also assesses whether adapted exergames for elderly the can be easy to use and effective enough to achieve good adherence to physical exercise and also improve the quality of life.

[3]http://www.iso.org/standard/66717.html

[4]http://www.android.com

Figure 2. Some of the physical exercises incorporated into the FFA platform [3]

In [9] it is proposed to use a game console for rehabilitation. The console is used to connect the therapist to the patient through a game. The patient has their care plan established during a first evaluation and during rehabilitation the plan is customized according to the performance and involvement in the game. First, the care plan is defined using the clinical information of the patient. Game sessions return information to the therapist who assesses whether the care plan is meeting the needs of the patient, and in the event that an adjustment is needed, a new care plan is created. The patient's involvement with the care plan is demonstrated by physiological data collected during the game session, such as their heart rate.

Jobs using PHDs such as [9] [8] deal with the integration of *ad hoc* form disregarding the adoption of a standard. Solutions that implement the X73-PHD standard, such as [7], explore adoption for other systems and not specifically for game engines. The main difference of our work compared to the previous ones is that we are focused on the integration of the PHDs to the games.

III. ISO/IEEE 11073 PHD

The X73-PHD standard, used in various research papers [10] [11], consists of a set of standards covering various aspects of medical data semantics. This includes a Domain Information Model (DIM) [12], nomenclatures, device specializations, device behavior, data transport, and communication protocol.

The X73-PHD standard defines that during a communication there are two elements, called Agent and Manager. The Agent is the communication side that produces the data, and can be exemplified as a scale, a glycometer, or a blood pressure meter. The Agent, being a personal electronic device and often wearable, must have some characteristics, such as low power consumption, to extend the autonomy of the device, and limited CPU, to decrease the size and weight of the device. Therefore, the communication protocol needs to be lightweight, avoiding long messages, and be efficient in terms of overhead, bandwidth, and CPU usage [13].

The Manager, on the other hand, is the communication side that receives the messages, in addition to interpreting them and making them available to any other system. It can be exemplified as a notebook, personal computer, smartphone, smartwatch, communication hub, etc. In addition to exchanging messages with the Agent, the Manager can send the data to another system. Sending data to the system can occur in a variety of ways, such as via interprocess communication, XML, JSON, and Sockets.

In order to assist in understanding and adopting the standard, Continua Health Alliance[5], a nonprofit organization of approximately 240 companies in the area of health and technology, is dedicated to reduce costs and improve results by developing technical guides for interoperability for systems using PHDs. In addition, Continua certifies products that fall within the standards specified in X73-PHD standard [14].

IV. Middleware Architecture

Figure 3 represents the proposed architecture, consisting of three parts, the PHD, the running game, and the Middleware, which runs on the same machine the game is running on. According to the X73-PHD standard, the PHD is the device that produces the data and can be defined as an Agent. The Middleware is part that receives the data and passes it to the game, defined as Manager.

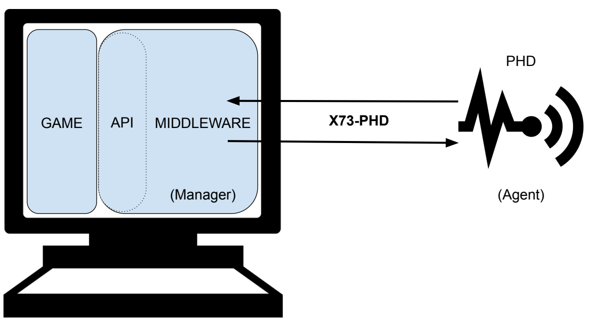

Figure 3. Middleware Architecture

The main function of the Middleware is to receive and pass on the data generated by the PHDs to the game. The Middleware also has other functions:

Data request: The developer requests a certain physiological data, and the middleware performs all the work of acquiring this data. Data acquisition considers fetching the device, pairing it, performing the measurement, and returning the data.

Continuous access to data: If the developer needs, the player's physiological data can be passed on to the game continuously, that is, the sensed data must be reported periodically at runtime.

Data history: The physiological data, in addition to being reported, can be stored. Thus, the developer can retrieve a set of data and send it to a clinical assessment service, for example.

PHD Simulation: The developer, at development time, uses commands to simulate PHDs, and can test his under-development game even without a real device.

Games must invoke API methods to initiate data exchange with the PHD and register for events that report their data. The API is a set of classes, methods, and events that can be invoked by the game. The data reported in the events follow a structure in JSON, widely adopted as an external representation of data in the communication between different applications.

V. Results

The Middleware is being built with the aid of the Continua Enabling Software Library (CESL)[6], which is publicly available in Continua Alliance's code repository. This library implements the technical details defined by the X73-PHD standard, such as the assembly of the data packet and the transport of these packets over different communication protocols.

The Middleware implementation is C++ based for Windows 32 and 64-bit platforms, where the vast majority of games are available.

As a way of demonstrating and validating the use of Middleware we have created a new plugin in Unreal Engine 4 (UE4), which adapts our Middleware for use within this game engine. This type of adaptation is made for several SDKs (Software Development Kits) present in UE4, such as Leap Motion SDK[7]. As in most of them the classes implement the Adapter software design pattern [15]. The function of this pattern is to convert one class to another that game engines expect.

In addition to adapting our Middleware to the game engine, the plugin uses particular features of UE4 to make programming blocks, called blueprints, which are available at a higher level to developers, exemplified in Figure 4. Thus, the developer can use all Middleware functionalities simply by creating blocks in the UE4 editor and developing their game.

[5]http://www.continuaalliance.org

[6]http://lnijira.atlassian.net/wiki/spaces/PCHA/overview

[7]http://developer.leapmotion.com/unreal

In Figure 4 we create a blueprint. The red top block "Event On PHDOutput" is the event registered to trigger every new data generated by the PHD. This event returns a JSON in text format, exemplified as "Output". Then we print this JSON on the screen using the top blue block called "Print String". This example is quite simple and is intended to demonstrate the basic steps of getting the data. From JSON the developer can retrieve all the data he needs for his game.

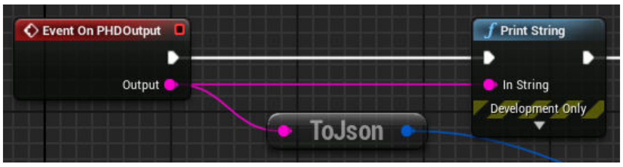

Figure 4. Printing the JSON with the data received from the PHD

VI. Conclusion and Future Work

During the literature review, a scenario was identified where PHDs are used to monitor patients during gaming sessions. This scenario was used as motivation to propose a solution that performs communication between the two main elements involved: game engine and PHDs.

The adaptation of the game engine can be replicated with the implementation of adapter classes in different game engines, simply adapting the API available. However, a plugin will be available to developers who want to use UE4. This way, one can take advantage of the elaborated architecture and proof of concept.

As a continuation of this work, still in progress, we hope to improve the Middleware and introduce new features. We will develop an exergame to validate the Middleware, and we will also evaluate the usability of the API through testing with game developers and evaluate performance, trying to find out if the Middleware operation impacts on important game metrics, such as the FPS (Frames Per Second).

References

[1] E. Brox, S. T. Konstantinidis, and G. Evertsen, "User-centered design of serious games for older adults following 3 years of experience with exergames for seniors: A study design," *JMIR Serious Games*, vol. 5, no. 1, p. e2, jan 2017.

[2] B. Ahmed, D. H. Kim, Y. Hwang, and S.-J. Park, "Treatment of alzheimer's, cognitive, chronic pain rehabilitation, depression and anxiety disorders in one system for elderly using VR," in *2018 15th International Conference on Ubiquitous Robots (UR)*. IEEE, jun 2018.

[3] E. I. Konstantinidis, A. S. Billis, C. A. Mouzakidis, V. I. Zilidou, P. E. Antoniou, and P. D. Bamidis, "Design, implementation, and wide pilot deployment of FitForAll: An easy to use exergaming platform improving physical fitness and life quality of senior citizens," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 1, pp. 189–200, jan 2016.

[4] V. S. Aguilar, C. Lamoth, N. Maurits, and J. Roerdink, "Assessing dynamic postural control during exergaming in older adults: A probabilistic approach," *Gait & Posture*, vol. 60, pp. 235–240, feb 2018.

[5] P. Cedillo, C. Sanchez, K. Campos, and A. Bermeo, "A systematic literature review on devices and systems for ambient assisted living: Solutions and trends from different user perspectives," in *2018 International Conference on eDemocracy & eGovernment (ICEDEG)*. IEEE, apr 2018.

[6] L. Niu, S. Saiki, and M. Nakamura, "Using non-intrusive environmental sensing for ADLS recognition in one-person household," *International Journal of Software Innovation*, vol. 6, no. 4, pp. 16–29, oct 2018.

[7] N. Georgi, A. Corvol, and R. L. B. Jeannes, "Middleware architecture for health sensors interoperability," *IEEE Access*, vol. 6, pp. 26 283–26 291, 2018.

[8] E. I. Konstantinidis, A. S. Billis, I. T. Paraskevopoulos, and P. D. Bamidis, "The interplay between IoT and serious games towards personalised healthcare," in *2017 9th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)*. IEEE, sep 2017.

[9] S. Ponte, E. Ferrara, S. Dellepiane, R. Ferretti, and S. Nardotto, "Technical concept of health data collection and integration data analysis for gaining meaningful medical information," in *Clinical Rehabilitation Experience Utilizing Serious Games*. Springer Fachmedien Wiesbaden, 2018, pp. 105–129.

[10] R. Lima, V. Ferreira, E. Caballero, C. Albuquerque, and D. Muchaluat-Saade, "An application layer proposal based on the standard ieee 11073 for simulation of digital health applications in wireless body networks," in *Proceedings of the 19th Brazilian Symposium on Computing Applied to Healthcare*. Niteroi, RJ, Brasil: SBC, 2019, pp. 234–245.

[11] H. Badawi, F. Laamarti, F. Arafsha, and A. El Saddik, "Standardizing a shoe insole based on ISO/IEEE 11073 personal health device (X73-PHD) standards," in *International Conference on Information Technology & Systems*. Springer, 2019, pp. 764–778.

[12] IEEE, "IEEE Health informatics–Personal health device communication - Part 20601: Application profile - Optimized Exchange Protocol," IEEE, Standard, 2014.

[13] J. Yao and S. Warren, "Applying the ISO/IEEE 11073 standards to wearable home health monitoring systems," *Journal of Clinical Monitoring and Computing*, vol. 19, no. 6, pp. 427–436, dec 2005.

[14] B. Piniewski, J. Muskens, L. Estevez, R. Carroll, and R. Cnossen, "Empowering healthcare patients with smart technology," *Computer*, vol. 43, no. 7, pp. 27–34, jul 2010.

[15] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design patterns: Abstraction and reuse of object-oriented design," in *Software Pioneers*. Springer Berlin Heidelberg, 2002, pp. 701–717.