Towards a Simulation-Based Medical Education Platform for PVSio-Web

Carlos Silva
Department of Informatics, University of Minho
& Center for Computer Graphics
Guimarães, Portugal
carlos.silva@ccg.pt

José Creissac Campos
Department of Informatics, University of Minho
& HASLab/INESC-TEC
Braga, Portugal
jose.campos@di.uminho.pt

Abstract— **Interface design flaws are often at the root cause of use errors in medical devices. Medical incidents are seldom reported, thus hindering the understanding of the incident contributing factors. Moreover, when dealing with a use error, both novices and expert users often blame themselves for insufficient knowledge rather than acknowledge deficiencies in the device. Simulation-Based Medical Education (SBME) platforms can provide appropriate training to professionals, especially if the right incentives to keep training are in place. In this paper, we present a new SBME, particularly targeted at training interaction with medical devices such as ventilators and infusion pumps. Our SBME functions as a game mode of the PVSio-web, a graphical environment for design, evaluation, and simulation of interactive (human-computer) systems. An analytical evaluation of our current implementation is provided, by comparing the features on our SBME with a set of requirements for game-based medical simulators retrieved from the literature. By being developed in a free, open source platform, our SBME is highly accessible and can be easily adapted to specific use cases, such a specific hospital with a defined set of medical devices.**

Keywords— **Human Factors, Medical Simulation, Safety Devices, Training, Presence, Usability, Computer Graphics, Video Games**

I. INTRODUCTION

Design problems and their contribution to use error have been acknowledged as a statistically relevant problem in medical devices incidents [1]. Walsh and Beatty [2], for example, estimated that 87% of medical incidents in patient monitoring contexts are due to human factors issues; Brixey, Johnson, and Zhang [3] estimated that use error has a greater impact in medical incidents than the impact caused by device failure. Moreover, medical devices incidents suffer from severe under-reporting, with the total of reports ranging from as low as 1.2% to 7.7% of the total number of occurrences [1].

Medical incident reports are often the base to understand the root causes of use error with medical devices. Nevertheless, increasing the rate of use error reporting is challenging due to a propensity to assign culpability to the user (often the one reporting the error) instead of assign culpability to the device (i.e. faulty interface design) or the work environment (i.e. user interruptions or inappropriate workload management) [4]. Strategies to mitigate use errors range from systematics analysis of the device design, in order to detect and eliminate potential use problems, to practitioner training (e.g., in simulation environments).

The application of automated reasoning techniques to the model-based analysis of medical devices has been a topic of active research in recent years [5][6][7]. Here, we are specifically concerned with the analysis of the user interfaces of the medical devices. To better integrate the formality and rigor of the tool-supported analysis, with the iterative and exploratory nature of a typical human-centered design process, the need for simulating the models has been identified [8]. PVSio-web [9], is a tool that support user interface prototypes generation from models of the user interface. The prototypes can be used to validate the design with stakeholders, interpret analysis results, but also as a more economical alternative to actual device usage during training. PVSio-web generated prototypes support interaction with the device, enabling the user to explore device behavior. The prototypes, however, are focused on specific devices, and do not provide a sense of the environment in which the device will be deployed.

Game-based learning is regarded as an emerging technology likely to have a large impact in training and education [10], and the use of simulators for skill training is widely regarded as beneficial [11][12]. Gamification is one way to foster engagement during simulation training of residents' surgeons and in some instances, surgeons are required to practice on a weekly-basis in Simulation-Based Medical Education (SBME) platforms [5]. Nevertheless, current SBME platforms are costly and targeted at highly specialized groups of practitioners, such as surgeons [13]. SBME platforms targeted at healthcare practitioners such as nurses are still scarce, despite the fact that these professionals have to deal with complex medical devices often prone to use errors (infusion pumps, ventilators, and radiology devices) [14] and, frequently, in hostile environments such as emergency rooms [15].

In this paper, we present the first steps in the development of a new SBME platform, based on PVSio-web. The goal is that the proposed SBME contextualizes interaction with medical devices, such as ventilators and infusion pumps, in a game-based simulation of a hospital environment. Thus, we add to the PVSio-web capabilities to simulate individual devices, the possibility to navigate a virtual environment representative of the space where the devices will be deployed.

The paper is organized as follows. In the next section, we will first set-up the requirements for developing an SBME, based on recently proposed design and development guidelines. Then, we will offer an overview of the proposed SBME and a description of its implementation. Finally, we will discuss the potentialities of SBMEs use for training of less specialized health care practitioners and the role that free, open-source platforms can have on fostering the use of such educational tools.

Post-print version of paper accepted for publication at ICGI 2018 - International Conference on Graphics and Interaction.

Publisher's version available at: http://dx.doi.org/10.1109/ITCGI.2018.8602845

II. REQUIREMENTS FOR THE DEVELOPMENT OF SBMES

SBMEs platforms exist in a wide variety of technologies, in different degrees of complexity, and for different application purposes [16]. Regarding base-technology, SBME can range from the paper-pen training toolkits such as the TeamSTEPPS [17], to virtual environments on desktop simulations (similar to computer games), high-fidelity immersive systems such as the training platform for the daVinci® surgical robot, or even real-environments on simulation centers resorting to real devices and surgical manikins. Regarding application, SBMEs might be designed to train professionals for a very specific procedure, such as a catheter insertion [18] or a knee replacement surgery [19]; to assess and score different professionals [20], to familiarize professionals with new medical devices before insertion on the ward; or even to foster team communication and other soft skills in different medical settings [21]. Despite all this variability, most SBMEs are applied as either a training or an educational tool, and this similarity in application scope led researchers on an effort to define requirements and key features of SBMEs.

Howe et al. [21] performed a review of work on SBMEs, and identified five key features for guiding the development of game-based simulation systems for medical training:

1. ***Automated Assessment*** – Use of scores or rating systems to evaluate users' performance. The assessment can be conducted through observational rating scales; self-assessments; or event-based approaches (scores systems implemented in the game);

2. ***Task Fidelity*** – Refers to the degree of similarity between tasks performed in the simulator and tasks performed on real equipment. Although the level of realism is something difficult to measure, the simulated environment should strive to mimic real-world interactions with objects, people, and teammates.

3. ***Interface Modality*** – Game-based simulators should allow for environment manipulation through user input. Some simulation events should be able to be controlled by the user and feedback of that interaction should be multimodal (sound, image, haptic feedback) and given timely.

4. ***Virtual Teammates*** – Simulations might encompass autonomous, animated agents that support face-to-face interactions. Virtual characters can serve as instructors guiding the narrative of the game, or playing roles such as team members, and thus promote the training of cooperation skills.

5. ***Customization and Adaptability*** – The training system platform must be flexible, customizable, and easy to use. Developers should be allowed to easily improve or expand previous implementations. This allows also for adaptations to each context of use by tailoring the simulation to represent the environment in a given hospital, or the use of specific equipment.

Other works are worth mentioning because they present similar and additional requirements for simulation environments. Issenberg et al. [16], presents a list of features of high-fidelity medical simulations, ordered by prevalence in the literature. It includes important additional features to Howe et al. features, such as:

6. ***Feedback*** – To foster learning, the systems should provide a way of informing users of their performance results. According to [16], educational feedback appears to slow decay of acquired knowledge while allowing learners to monitor their progress toward skill acquisition.

7. ***Multiple Learning Strategies*** – The SBME should be adaptable enough to be used in different educational contexts, such as instructor–centered education (lectures, workshops, educational videos, tutorials) or individual, independent learning without instructor.

8. ***Capture Clinical Variation*** – High-fidelity medical simulators that can capture or represent a wide variety of scenarios are more useful and allow for additional educational and training valences.

9. ***Individualized Learning*** – According to [16], the SBME should convey the opportunity for learners to have reproducible, standardized educational experiences where they are active participants, not passive bystanders.

Taken together, these two lists offer a comprehensive set of nine requirements that should be considered when developing new SBMEs.

We now analyse PVSio-web prototypes against these requirements:

1. Not being designed as a SBME platform, PVSio-web naturally does not support **automated assessment**.
2. Being based on models of the device, prototypes provide a high level of **task fidelity** at the device interaction level; however, they lack a more holistic view of the device in its environment.
3. Prototypes support user interaction and can support multiple **interface modalities**; e.g. voice output is supported.
4. Being focused on user-device interaction, PVSio-web prototypes currently lack support for **virtual teammates**.
5. The fact that prototypes are based on models makes them highly **customizable and adaptable**.
6. As with requirement 1, the prototypes were not designed with SBMEs in mind, so educational **feedback** is not directly provided; however, feedback about the interaction is provided as it would be in the real system.
7. Prototypes can be deployed in a number of ways so **multiple learning strategies** are supported and have indeed been experimented with.

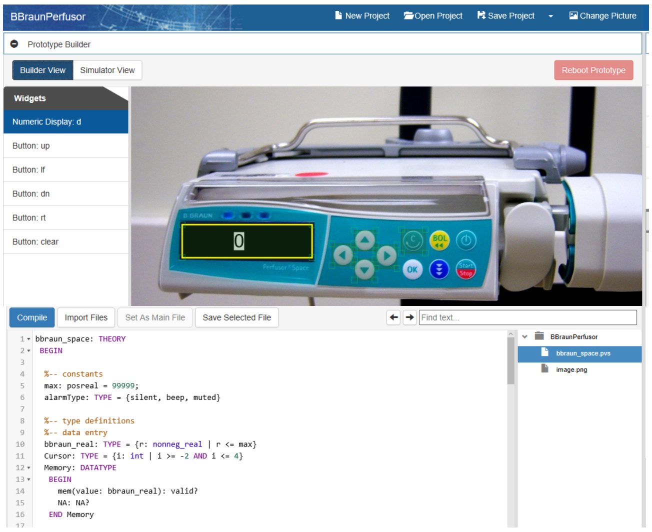
8. Device prototypes are high fidelity and can **capture clinical variation**.
9. Prototypes are focused on user device interaction, so participants become active participants thus providing the basis for **individualized learning**.

In summary, two sets of features are needed to create SBMEs with PVSio-web: (1) Learning support features related to requirements 1 (Automated Assessment) and 6 (Feedback); (2) Features related to providing a more immersive experience of the clinical environment where the device (or devices) will be used in practice. This last set of features relates mainly to requirements 2 (Task Fidelity) and 4 (Virtual Teammates). Herein, we focus on the second set of features. We will do this by developing a virtual environment where PVSio-web prototypes will be deployed. This will add to the already realistic device prototypes a sense of the physical space where they situated, without the need of using the actual physical space, thus improving task fidelity at a low cost. At the same time, it will also enable support for virtual teammates.

In the next sections, we will offer an overview of the proposed SBME and a description of its implementation and, in Section V, we will evaluate how well our proposed SBME complies with this set of nine requirements.

III. PVSIO-WEB

The proposed SBME is composed of a game module that provides a structured, contextualized, and realistic use environment for the PVSio-web tool **Error! Reference source not found.** – a graphical environment for facilitating the design and evaluation of interactive (human-computer) systems (see Figure 1). PVSio-web can be used to define, generate, and evaluate realistic interactive devices (existing devices or new concepts/prototypes) from formal methods. Since its release (in 2013), it has been mainly used to model and evaluate medical devices using formal methods, and to create training material for device developers and users in the medical domain – such as infusion pumps, and ventilators.


```
bbraun_space: THEORY
 BEGIN

  %-- constants
  max: posreal = 99999;
  alarmType: TYPE = {silent, beep, muted}

  %-- type definitions
  %-- data entry
  bbraun_real: TYPE = {r: nonneg_real | r <= max}
  Cursor: TYPE = {i: int | i >= -2 AND i <= 4}
  Memory: DATATYPE
   BEGIN
     mem(value: bbraun_real): valid?
     NA: NA?
   END Memory
```

Figure 1: Screenshots of the main tools provided by the PVSio-web environment. Top image – Prototype builder; Bottom image - model Editor and a model snippet generated from emuchart editors.

As stated in **Error! Reference source not found.**, state-of-the-art verification tools like PVS generally have minimal front-ends that create barriers for use by multidisciplinary teams and difficult engagement of non-experts. PVSio-web was proposed as a way of reducing these barriers and provide an interface to modeled medical devices.

Our present work intends to take the tool a step further towards functioning as a SBME. We achieve this by developing a game that contextualizes and facilitates access to models available on the PVSio-web. Hence, the SBME we propose herein might be regarded as a *game mode* of the PVSio-web tool. At any given points in the game it calls upon interfaces prototyped in PVSio-web 2.3.

More precisely, the SBME is intended to provide a game-environment where navigation through different rooms of a hospital and interaction with different medical devices is possible. There is no pre-defined game narrative, thus free exploration of the game environment is possible and interaction with the different devices might follow any order. This environment is intended for external observers (evaluators or educators) to define their own script of interaction evaluation.

IV. METHODOLOGY

A. *Simulation Software*

To implement our SBME, we used Blender (release 2.78c) [22], an open-source 3D computer graphics software, written in C, C++, and Python, with an integrated game engine called Blender Game. By choosing Blender as our simulation software, we are able to guarantee free, open-source distribution of a cross-platform SBME. Thus, other developers might be able to easily extend or adapt the game to each particular use scenario (i.e., different hospitals might require simulation of different medical devices). Moreover, by using free, open-source software we are keeping in line with the position adopted in the development of the PVSio-web tool.

B. *Implemented environment*

Figure 2 presents the game flowchart and the adopted relation between the game mode and the PSVio-web tool in order to function as a SBME for training on medical device use.

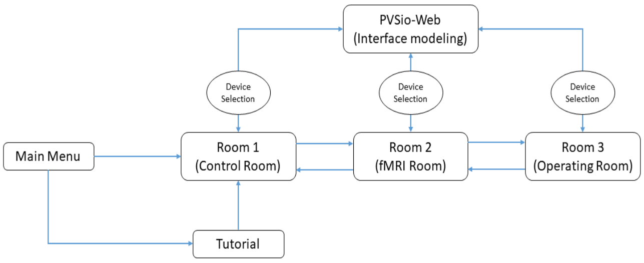


Figure 2: Game flowchart.

The scenario for the game mode (see also Figure 3) is a hospital ward, more specifically three adjoining rooms composed of a control room (Room 1), an fMRI (functional Magnetic Resonance Imaging) room (Room 2), and an operating room (Room 3). The player controls a character (a nurse) that can move between rooms and select the different devices available in the graphical environment in order to interact with them. By default the player starts at room 1, a control room for the fMRI machine. By opening the door with the fMRI machine off, the player can move to room 2, where it can view the interface of fMRI machines. By moving into room 3, the player enters an operating room where he or she may interact with several medical devices, such as infusion pumps and ventilators.

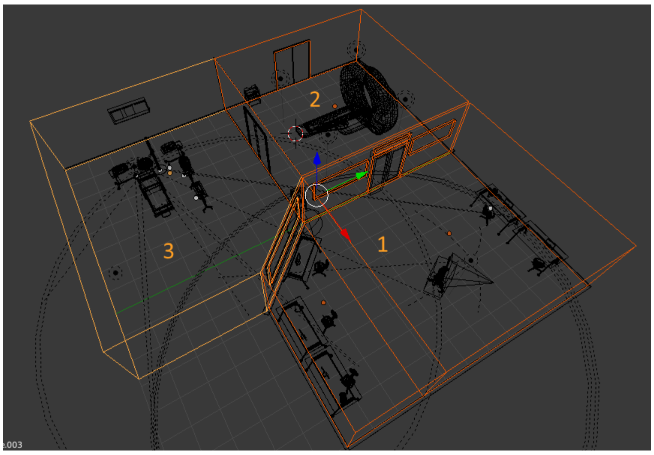
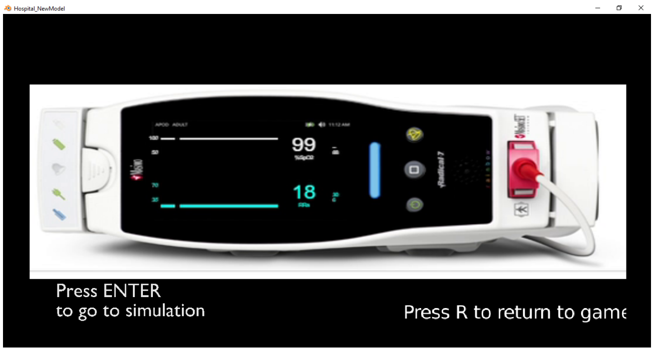


Figure 3: Game Scenario. Consists in three rooms.

The transition between the game environment and the PVSio-web tool simulation is activated upon the selection of a specific device in a specific hospital room. Each time a player chooses to interact with a given medical device, the actuation on the main scenario stays on hold and a close-up of the medical device interface shows-up on the screen (Figure 4).



Press ENTER
to go to simulation

Press R to return to game

Figure 4: Close-up of the Massimo Radical 7® Pulse Co-Oximeter® a monitoring device that keeps track of patients' oxygen saturation.

To provide an environment familiar with a hospital ward, the rooms where modeled using high polygon count objects and high definition textures (Figure 5). The current total polygon count is 238 615.

Next, we will discuss the main aspects of the environment. Namely, the available commands, the logging that is done, and concrete medical devices that were included.

1) Commands. The game features a tutorial explaining the commands available to interact with the environment. Namely, commands for character navigation control and commands for interaction with elements from the graphical environment. Character navigation is performed using W-A-S-D for up-left-down-right navigation respectively, and by using the mouse for camera pan and tilt control (resorting to the *look mode* on Blender's Game mouse actuator). Character control is implemented using the game logic mode of Blender Game.

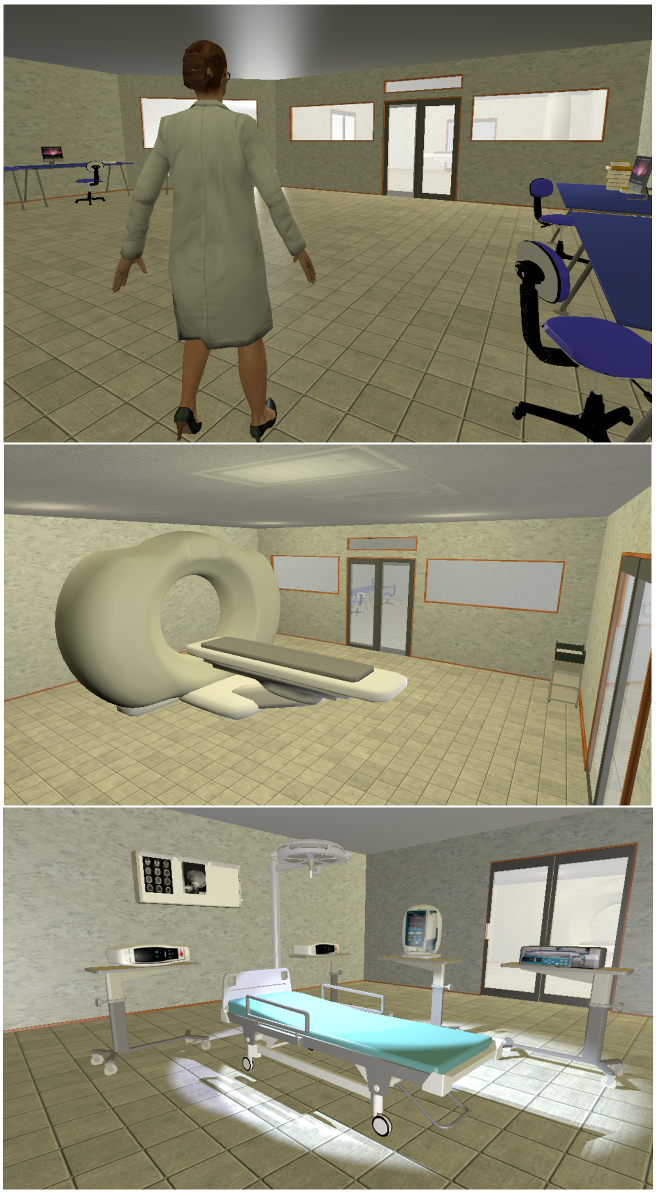
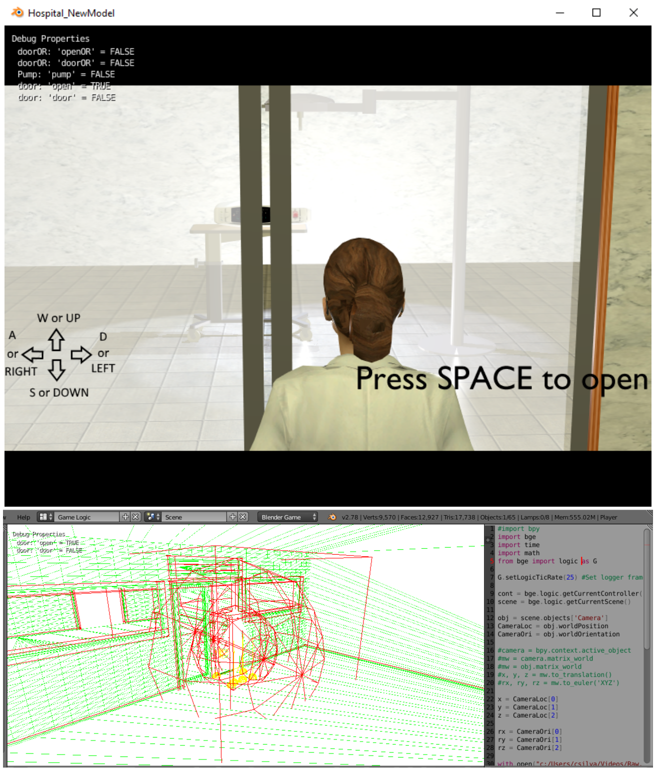


Figure 5: Screenshots from the game environment. Top-image is the control room; Midle image is the fMRI room; and bottom image is the Operating Room.

Interaction with elements from the graphical environment were implemented resorting to Blender's *near sensors* and Boolean properties controlling animations (e.g. opening doors) or passage to new scenes (e.g. Medical devices close-ups). Figure 6, shows, through physics visualization, how the near sensor operates. A spherical area around the character is defined on the near sensor by setting a radius value to trigger the sensor (inward sphere) and a radius value to reset the sensor (outward sphere).

Interactions on the SBME are restricted to opening of doors for room-to-room, character navigation, entering into close-up visualization of the different medical devices, and re-direction for the PVSio-web landing page of each device available in the SBME. Most of the interaction was programmed as a game logic in Blender Game. Figure 7, shows a simplified version of the game logic for the coordination between the *near sensor* and the *open door* interaction. A Boolean property is controlling the animation openDoorOR. A near sensor triggers the instructions in an overlay scene and sets the openOR property to true (if "SPACE" is pressed within the near sensor distance) allowing to run the animation of the door opening.



```
Debug Properties
 doorOR: 'openOR' = FALSE
 doorOR: 'doorOR' = FALSE
 Pump: 'pump' = FALSE
 door: 'open' = TRUE
 door: 'door' = FALSE
```

```
#import bpy
import bge
import time
import math
from bge import logic as G

G.setLogicTicRate(25) #Set logger fram

cont = bge.logic.getCurrentController(
scene = bge.logic.getCurrentScene()

obj = scene.objects['Camera']
CameraLoc = obj.worldPosition
CameraOri = obj.worldOrientation

#camera = bpy.context.active_object
#mw = camera.matrix_world
#mw = obj.matrix_world
#x, y, z = mw.to_translation()
#rx, ry, rz = mw.to_euler('XYZ')

x = CameraLoc[0]
y = CameraLoc[1]
z = CameraLoc[2]

rx = CameraOri[0]
ry = CameraOri[1]
rz = CameraOri[2]

with open("c:/Users/csilva/Videos/Raw
```

Figure 6: Top image, interface shown during the game in order to prompt the player to open the door. Down image, physical visualization of the near sensor in Blender Game.

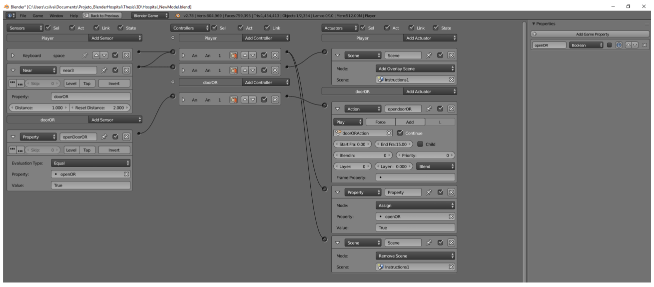


Figure 7: A simplified Game Logic example for an opening door action.

2) Logs. For possible future analysis of the interaction and game performance, the environment is currently logging the character position throughout the entire duration of the game. This data might be interesting for analysis of the player's exploration patterns, or to quantify the amount of time the player spends in interaction with each device. The logger is implemented in Python through Blender's scripting platform. At each cycle of the defined logging frequency, the camera global location and orientation is printed-out in a .csv file.

Dumping of the gameplay is also included as an option. By resorting to the *Rasterizer library* and using the *.makeScreenshot( )* function, we allowed gameplay dumping at a user specified frame-rate.

3) Medical Devices. The current version of the environment includes the medical devices described in Figure 8. Each device was modeled in Blender 2.78c (including the fMRI machine). These devices are shown in room 3 (see Figure 5). Once the player is near a device, an overlay scene instructs the player to press 'I' to view the device in a close-up mode (Figure 4). The close-up view shows a video of the device in running (for the pulse co-oximeter) or waiting mode (for the infusion pumps). This effect is achieved by UV-mapping an *.mp4* video as a texture of the object in the close-up scene. The game logic of that object uses an always sensor to run a python script that loads and refresh the specified video into the object's texture, as follows:

```
IF "video" exists in the selected object
        "video" EQUALS to object's "video"
        REFRESH "video"
ELSE
        READ object's material and texture
        SET "video" variable as object's video texture
        "video" source EQUALS to movie location
        IF "video" Loop parameter is false, make it true
        PLAY "video"
```

If the user, chooses to press "ENTER", when in close-up mode, the game is paused and a web-browser running that medical device simulation in PVSio-web is shown in full-screen. Inclusion of the PVSio-web simulation in the Blender game engine was not possible, since currently Blender does not support rendering of HTML inside a modal platform-native window as, for instance, Unity does (see Project Awesomium [23] for Unity). Keeping our SBME open-source was a requirement, thus we kept Blender as the development platform even though we have to deal with the limitation of having to open a web-browser in order to interact with the PVSio-web simulation

Figure 8: Medical devices currently modeled and available in our SBME. The top-left one is the *Massimo Radical 7® Pulse Co-Oximeter®* a monitoring device that keeps track of patients oxygen saturation, pulse rate, and perforation index; the bottom-left one is the *BBraun® Infusionmat IV* Pump, a programmable syringe based infusion pump; the right one is the *Alaris® GP Volumetric Pump* an infusion pump suited for drugs therapy, blood transfusions, and parenteral feeding

This project can be downloaded at:
https://github.com/Carlos-CCG/HospitalBlender

V. EVALUATION

The current state of our SBME is already compliant with some of the requirements listed in Section II. Nevertheless, additional development is needed in order to fulfill all the nine requirements deemed relevant for game-based simulators in healthcare. We now consider the current state of development against the nine identified requirements:

1. ***Automated Assessment*** – Users' performance when interacting with the medical devices is not directly recorded by the SBME. Automated assessment could be possible if included in the models of the devices. This however is far from ideal. Currently, it is possible to evaluate the success of tasks completion through external observation or by resorting to interaction recording software for usability assessment, such as the OvoSolo® or the ActivePresenter7 software. In this sense, the SBME allows for *post-hoc* assessment of skilled interaction behavior. Future developments should consider the automation of the assessment process inside the game engine. As a side note, *post-hoc* assessment of the interaction with prototypes will be useful for usability assessment with end-users, during the development process of new medical devices

2. ***Task Fidelity*** – By resorting to PVSio-web models our SBME allows interaction with medical devices simulations that mimic the exact behavior of real medical devices. Thus, learning transfer should be possible as the user might be prompt to perform tasks which are representative of the real-world scenarios. This is particularly important for training and assessment of performance on tasks that are prone to interaction errors (e.g. data entry in infusion pumps) or that highlight real faulty device behavior. Results of this interaction might serve, for instance, to raise healthcare professionals' awareness of the most probable use errors in each medical device. Additionally, the simulation now supports a sense of the environment where the devices are used and has the potential for implementing multi-user scenarios.

3. ***Interface Modality*** – Multiple modalities in the interaction with the simulated environment are possible. Nevertheless, the current implementation of the environment is still unimodal, lacking in auditory stimuli (unlike the individual device prototypes). Future developments should consider a multimodal simulation environment.

4. ***Virtual teammates*** – Although the environment supports developing multiple avatars, which might be controlled programmatically or by other users of the environment in a multi-user setting, currently there are no virtual characters serving as instructors or playing other relevant role in the scope of the developed environment. This feature might be relevant for future applications intended to approach training or education in more complex settings, such as the ones making use cooperative work and medical cyber-physical systems.

5. ***Customization and Adaptability*** – By being developed in an open-source framework, our SBME allows customization and expansion by other developers. Moreover, recent developments in PVSio-Web facilitate the implementation of medical device simulations by non-experts in formal modelling [24].

6. ***Feedback*** – Due to the nature of its implementation, feedback on the interaction during a simulation trial is only possible if it is part of the formal description of the medical device simulation in PVSio-web. If the PVS model includes warnings, error messages, and state indicators, those will be visible during a simulation trial. Nevertheless, as the medical device simulation is developed outside Blender, this feedback behavior has to be implemented in PVSio-web. Additional educational feedback should be included in the game environment, but is dependent on the implementation of automated assessment (requirement 1).

7. ***Multiple Learning Strategies*** – Due to its nature as a platform for free interaction with the displayed medical devices, our SBME might be used in significantly different environments or applications. It can be used to develop educational videos illustrating the interaction problems that faulty implementations might create, to conduct controlled interaction assessment of healthcare professionals, or simple to provide a way for healthcare professionals to get familiar with the interface of a new medical device before implementation in the ward. Adding the simulated environment, significantly increases the flexibility to implement different learning strategies.

8. ***Capture Clinical Variation*** – By being able to provide an interaction simulation with different medical devices, our SBME is able to account for some of the variability that healthcare professionals might encounter in real-world scenarios. Again, the simulated environment, significantly increases the expressive power of the tool.

9. ***Individualized Learning*** – By allowing reproducibility of scenarios of interaction and standardize educational experiences, and by demanding the users to be active participant during the simulation, our SBME might be applied in contexts of individual learning.

Table 1 presents a summary of the above discussion on the current state of the proposal as well as a comparison with the original PVSio-web prototypes, when analyzed against the requirements. The main conclusions is that support for requirements 2, 4, 7 and 8 has been improved. It should also be considered that the basis are established for improvements in the other requirements.

Table 1. Assessment

Req.	1	2	3	4	5	6	7	8	9
PVSio-web	×	~	~	×	✓	~	~	~	✓
Proposal	×	↗	~	~	✓	~	↗	↗	✓

× - Not satisfied; ~ - Partially satisfied; ↗ - Improved; ✓ - Satisfied.

VI. FUTURE WORK

Although some assessment is already feasible, the SBME should be expanded with a well-developed module for automated assessment of performance. By simulating contexts with standardized tasks, in which a particular action has to be performed in a particular device, the automated scoring of users' performance will be facilitated. As it is, the SBME allows only free exploration of user-interface interaction, requiring effort on external observers in order to keep track of users' performance. Nevertheless, a game logic with specific tasks and classifiable outcomes can be developed and interaction performance can be better assessed in future developments.

One of the main currently advantages of our SBME, the possibility to customize and adapt the simulation, might be further enhanced by developing structured ways to conduct such customization. One way of doing so, is to control the process of adding new medical devices. Currently, this process is time consuming. The developer has to: (1) implement a formal description and a model for simulation in PVSio-web and, (2) create a 3D model in Blender with the game logic, which allows to player to see the device in close-up mode and progress to the landing page of the device simulation in PVSio-web. Currently, part (2) is already implemented as we offer an assets library in Blender, with a preview module for each asset, which allows the SBME scenario creator to easily select previously developed models of new medical devices.

VII. DISCUSSION

In this paper we present the basis for a free, open source, SBME for training of user-interface interaction in medical devices. Typical SBMEs are expensive, difficult to adapt to new use cases, and targeted at a very specific sets of medical specialties [8]. The current state of our SBME provides the basis for developing training programs targeted at specific use cases (i.e. an hospital and their specific device models), and for less-specialized healthcare professionals, such as nurses. This particular group of healthcare professionals has to interact with an increasingly different set of medical devices, sometimes having to work in the same hospital with different models or brands of equipment targeted for the same function. All of this variability in medical device use hardens the process of learning and makes training even a more important part of the job. In a 2015 survey [15], 76% of respondents (from a pool of 113 resident nurses) stated the need of continued practical training. An expandable and easily adaptable SBME might configure a possible solution for this acknowledge necessity.

ACKNOWLEDGMENT

This work was supported by project "NORTE-01-0145-FEDER-000016" financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF). The authors wish to thank Paolo Masci, developer of PVSio-web, for extensive support and input.

REFERENCES

[1] Vicente, K. J., Kada-Bekhaled, K., Hillel, G., Cassano, A., Orser, B. A., 2003. Programming errors contribute to death from patient-controlled analgesia: case report and estimate of probability. *Canadian Journal of Anesthesia* 50 (4), 328.

[2] Walsh, T., Beatty, P., 2002. Human factors error and patient monitoring. *Physiological measurement* 23 (3), R111

[3] Brixey, J., Johnson, T. R., Zhang, J., 2002. Evaluating a medical error taxonomy. In: Proceedings of the AMIA Symposium. American Medical Informatics Association, p. 71.

[4] Martin, Jennifer L., et al. "Medical device development: The challenge for ergonomics." *Applied ergonomics* 39.3 (2008): 271-283.

[5] B. Kim et al., "Safety-assured development of the GPCA infusion pump software," in Proceedings of the 9th ACM International Conference on Embedded Software. New York, NY, USA: ACM, 2011, pp. 155–164. DOI: 10.1145/2038642.2038667

[6] M. L. Bolton and E. J. Bass, "Generating erroneous human behavior from strategic knowledge in task models and evaluating its impact on system safety with model checking," IEEE Trans. Syst., Man, Cybern., Syst., vol. 43, no. 6, pp. 1314–1327, 2013.

[7] M.D. Harrison, M. Drinnan, J.C. Campos, P. Masci, L. Freitas, C. di Maria and M. Whitaker (2017) "Safety analysis of software components of a dialysis machine using model checking". In Formal Aspects of Component Software, volume 10487 of Lecture Notes in Computer Science, pages 137-154. Springer.

[8] M.D. Harrison, P. Masci, J.C. Campos and P. Curzon (2017) "Verification of User Interface Software: the Example of Use-Related Safety Requirements and Programmable Medical Devices". IEEE Transactions on Human-Machine Systems, 47(6):834-846, December.

[9] Masci, Paolo, et al. "PVSio-web 2.0: Joining PVS to HCI." *International Conference on Computer Aided Verification*. Springer, Cham, 2015.

[10] Kerfoot, B. Price, and Nicole Kissane. "The use of gamification to boost residents' engagement in simulation training." *JAMA surgery* 149.11 (2014): 1208-1209.

[11] Dawe, S. R., et al. "Systematic review of skills transfer after surgical simulation‐based training." *British Journal of Surgery*101.9 (2014): 1063-1076.

[12] Cook, D. A., Hatala, R., Brydges, R., Zendejas, B., Szostek, J. H., Wang, A. T., & Hamstra, S. J. (2011). Technology-enhanced simulation for health professions education: a systematic review and meta-analysis. *Jama*, 306(9), 978-988.

[13] Zendejas, Benjamin, et al. "Cost: the missing outcome in simulation-based medical education research: a systematic review." *Surgery* 153.2 (2013): 160-176.

[14] C. Silva, P. Masci, Y. Zhang, P. Jones and J.C. Campos A Use Error Taxonomy for Improving Human-Machine Interface

Design in Medical Devices. *ACM SIGBED Review (Special Issue on Medical Cyber Physical Systems - MCPS 2018)*. (to appear)

[15] Ewertsson, Mona, et al. "Use of technical skills and medical devices among new registered nurses: A questionnaire study." *Nurse education today* 35.12 (2015): 1169-1174.

[16] Barry Issenberg, S., et al. "Features and uses of high-fidelity medical simulations that lead to effective learning: a BEME systematic review." *Medical teacher* 27.1 (2005): 10-28.

[17] Clancy, C. M., & Tornberg, D. N. (2007). TeamSTEPPS: assuring optimal teamwork in clinical settings. *American Journal of Medical Quality*, 22(3), 214-217.

[18] Barsuk JH, McGaghie WC, Cohen ER, et al. Use of simulation-based mastery learning to improve the quality of central venous catheter placement in a medical intensive care unit. *J Hosp Med*. 2009;4:397–403.

[19] Jun, Yongtae, et al. "Anatomic basis 3-D surgical simulation system for custom fit knee replacement." *International Journal of Precision Engineering and Manufacturing* 13.5 (2012): 709-715.

[20] LeBlanc, Justin, et al. "A comparison of orthopaedic resident performance on surgical fixation of an ulnar fracture using virtual reality and synthetic models." *JBJS* 95.9 (2013): e60.

[21] Howe, J., Puthumana, J., Hoffman, D., Kowalski, R., Weldon, D., Miller, K., ... & Ratwani, R. (2018, June). Development of Virtual Simulations for Medical Team Training: An Evaluation of Key Features. In Proceedings of the International Symposium on Human Factors and Ergonomics in Health Care (Vol. 7, No. 1, pp. 261-266).

[22] Blender Foundation - https://www.blender.org/foundation/

[23] Awsomium - https://assetstore.unity.com/packages/tools/gui/in-app-web-browser-57532

[24] Watson, Nathaniel, Reeves, Steve, & Masci, Paolo, 2018. “Integrating user design and formal models within PVSio-Web”. (In Press)