Video games as inspiration for scientific software

Francisco Queiroz[1,2][*]   Rejane Spitz[1] [†]   Pablo Elias[2] [‡]   Rodrigo Pinheiro[2] [§]   Thiago Azevedo[2] [¶]
Vinicius Rodrigues[2] [‖]   Luciano Reis[3] [**]   Alberto Raposo[2] [††]

[1]Arts & Design Dept., Pontifical Catholic University of Rio de Janeiro, Brazil
[2]Tecgraf Institute, Pontifical Catholic University of Rio de Janeiro, Brazil
[3]Petrobras Research Center, Brazil

ABSTRACT

This paper presents a study on challenges and opportunities for the gamification of scientific software, with the purpose of investigating methods for the transposition of design elements from games into that particular type of software. This investigation was based on action-research methodology, reflecting on the collaboration between participants in addressing practical issues, as well as inferring theoretical insight from that experience. Based on deliberations about low and high-fidelity prototypes, the research suggests that design elements from games could improve usability and insight, if adapted to scientific software specific needs and, more importantly, adequate to its development environment. We propose that video game design elements have the potential to improve scientific software, as long as implementation is feasible – in which case we recommend a conservative mindset during design stages, informed by a clear understanding of system requirements, development time, and human resources.

Keywords: Gamification, scientific software, game design, computational science

1 INTRODUCTION

Gamification is the use of game design elements in non-gaming contexts [10]. It has been applied in many areas as a means to generate compelling experiences, often approaching usability and HCI issues from a game design perspective [9], [11], [33]. Extensively and successfully used in citizen science [29] and science education [28], gamification could potentially improve scientific software usability [48], which is often neglected [36], [38], [2]. However, scientific software can be a unique environment regarding software use and development, subject to specific needs and challenges that could potentially impact on gamification processes. To better investigate potential gains and possible implications of gamifying scientific software, we have conducted a research to identify opportunities and methods for improvement, hopefully finding ways to address usability shortcomings.

This study took place at the Tecgraf Institute, where a team of five developers and one part-time designer have been developing SiVIEP for the Petrobras company since 2010. SiVIEP is a software designed for the visualization of exploration and production 3D models and simulation results in oil & gas projects, which can be used in both desktop PCs and Virtual Reality (VR) environments. SiVIEP is currently part of a broader research on a novel, more integrated approach to flow simulations within the oil & gas industry. During a four-month period, that group was joined by a doctoral researcher in design investigating gamification of scientific software. The researcher in question is also a member of the Tecgraf Institute – although allocated in a different team. The study was performed as part of a doctoral research and undertaken for accreditation in an HCI-related discipline taught by the coordinator of the group developing SiVIEP. As the software's development is based on Scrum methodology, the research also counted on the occasional participation of Product Owners (POs) – stakeholders who establish product's priorities – a similar role to Project Manager, although more involved with development decisions [8]. In SiVIEP's case, POs are not based in the same location as the development team.

Throughout the next sections of this article, the research is described as follows: Related Work discusses similar projects and papers; Methodology presents a brief overview of the action-research methodology upon which the study was based; Research Development Overview discusses the evolution of the study as it progressed; Lessons Learned presents findings and potential contribution for the area of study. Finally, the paper is summarized in the Conclusion section.

2 RELATED WORK

Gamification is characterized by the use of game design elements in contexts that are external to games [10], often making use of such elements (e.g.: points, scoreboards, feedback, story/theme, etc.) as motivational affordances capable of giving activities a game-like structure and feel [12]. Methods for gamifying software by taking inspiration from games have been successfully developed by Popa – who proposed gamification as the "use of cross-media references from games" to other media [32, p. 17], and by Deterding, who has elaborated a method of "translat[ing] game design insights for interaction design" [9, p. 329]. On a related topic, Souza and Souto have proposed the evaluation of gamified applications from the perspective of gameplay heuristics [40] – a method that could help measure the depth and effectiveness of the gamification process. Wolff acknowledged gamification's potential to improve usability, proposing its inclusion in a discipline about scientific software development [48].

Scientific software can be defined, according to Kelly [19], by three main characteristics: it is developed to answer a scientific question; it relies on the close involvement of an expert in its scientific domain; and it provides data to be examined by the person who will answer the scientific question at hand. The subject of scientific software usability has been addressed by several authors. Ranging from general guidelines to particular case studies, these articles unanimously emphasize the need for understanding not only the scientific work at hand, but also how scientists work. This impression was highlighted by Springmeyer [41], Javahery et al. [15], and Pancake, who has elaborated a four-step design methodology for scientific software design [31]. Keefe has discussed unique charac-

---

[*]e-mail:chico@puc-rio.br
[†]e-mail:rejane@puc-rio.br
[‡]e-mail: pelias@tecgraf.puc-rio.br
[§]e-mail:rbraga@tecgraf.puc-rio.br
[¶]e-mail: thiagom@tecgraf.puc-rio.br
[‖]e-mail: viniciuslr@tecgraf.puc-rio.br
[**]e-mail: luciano.reis@petrobras.com.br
[††]e-mail:abraposo@tecgraf.puc-rio.br

teristics such as complex data, need for precision and specialized, motivated users [16]. A number of case studies for scientific software interface design corroborated such claims, and some have extensively reported the design process of user interfaces, such as the ones for *Omero* [24], [39], [23] and the *Enzyme Portal* [6]. More recently, Thomer et al. have explored participatory design by organizing a hackaton where scientists and designers collaborated, generating ideas for a user-centered interface for taxonomy software [44].

Initiatives for gamifying scientific software have been particularly prolific in the field of engineering by taking advantage of video games' immersive qualities. Kosmandoudi et al. discussed how game mechanics, aesthetics and technologies have been applied to BIM and CAD [20]. The subject was previously addressed by Boeykens, who highlighted how the use of video game engines, unlike traditional tools, enabled the construction of highly interactive and compelling experiences [5]. The use of video game technology in engineering was also discussed by Moloney [27], for whom new prototyping practices might demand a paradigm shift in architecture and engineering education. This topic was also approached by Keenaghan and Horvath, who discussed the potential of gamified virtual and augmented reality as learning platforms for built environment education [17].

Gamification initiatives in engineering are not exclusively based on immersive traits. Autodesk research team has developed a gamified tutorial system for *AutoCAD* that makes use of motivational affordances such as goals, feedback, rewards, progression, story, and graphics [21]. Researchers from the Chinese University of Hong Kong have developed a gamified application for promoting collaboration between developers and potential buyers in mass housing production [22]. Researchers from the University of Calgary have been applying gamification to engineering education, including the development, by students, of games designed to illustrate and simulate topics presented in class [26].

The present study tries to bridge those subjects, discussing approaches for applying design elements from games in engineering software, with emphasis on issues and challenges typically associated with scientific software development.

3 METHODOLOGY

Aiming at empirically-based, first-hand experience with our subject, we have chosen the action-research methodology as a basis for our investigation. Such methodology is often employed when innovation and improvement of current practices are desirable. Supported by bibliographic research and theoretical frameworks, action-research allows knowledge to be inferred through the problematization and confrontation of practical, localized issues. In this particular case, our goal was gaining an understanding on challenges and opportunities for gamifying scientific software by addressing usability issues eventually found on SiVIEP. Figure 1, built on top of David Tripp's representation of the action-inquiry cycle [45], illustrates the overall process.

The research took approximately four months. During that period, a total of nine meetings were held – approximately one meeting every two weeks. Meetings were used for identifying problems, debating and proposing possible solutions, brainstorm sessions, evaluating proposed ideas, and planning next steps. In-between meetings, the reseacher would develop a prototype to propose features reflecting identified needs. Occasionally, meetings could feature debates on game design practices and theories underlying the research. In one occasion, a semi-structured interview took place. After meetings, participants would receive PDF documents containing meetings minutes summarizing discussed points.

4 RESEARCH DEVELOPMENT OVERVIEW

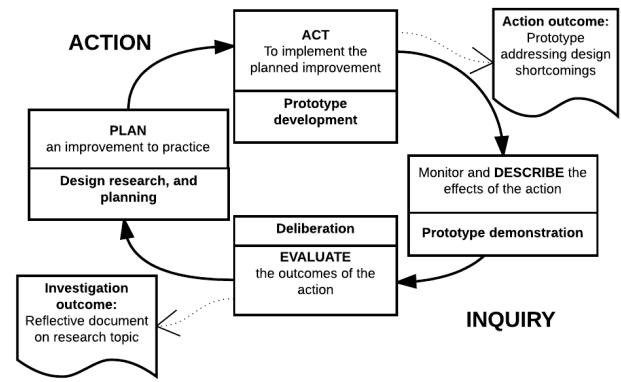In this section, we present an overview of the research development.

Figure 1: Research process overview.

4.1 Stage 1: problem definition

The first three meetings were used in the identification of usability issues that could be addressed by gamification.

4.1.1 First meeting: software overview and perceived shortcomings

The first meeting between researcher and developers was dedicated to an overview of the project, including development and usability issues. SiVIEP was presented as a software for visualizing integrated simulation of oil & gas projects, which incorporated data from diverse sources (production facilities, reservoirs, process plants, etc.) into a single virtual, three-dimensional representation of a production field [34]. A screenshot from the software can be seen in Figure 2.
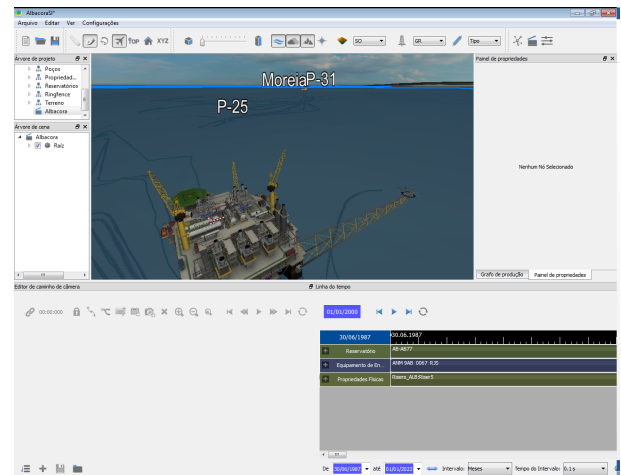

Figure 2: SiVIEP screenshot.

Besides first-person navigation around that virtual environment, SiVIEP features supportive functionalities for data visualization, such as a *Timeline* – which allows users to visualize the evolution of project through time, and a *Graph* – which displays a comprehensive, schematic view of the interconnected equipments. Both functionalities can be seen in Figure 3.

The main use of the software had been, until then, VR exhibitions of oil fields. It was noted that the VR mode was lacking features found in the desktop version, most notably the *graph* and *timeline* windows that allowed for time control and schematic views of represented environments. Additionally, it was expressed the desire
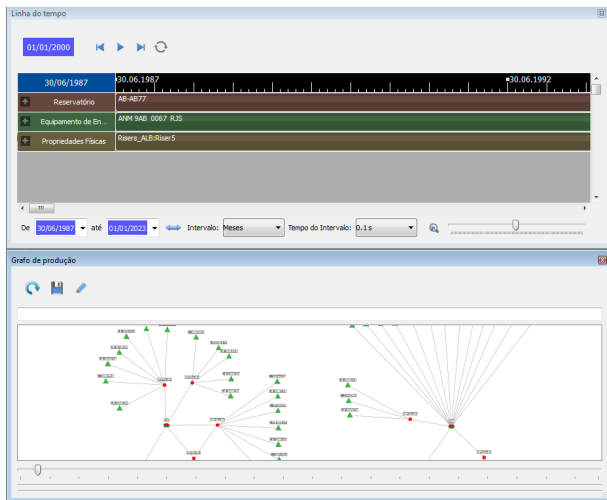
Figure 3: Timeline (top) and Graph (bottom) views.

to unify desktop and VR interfaces – thus avoiding need for separate feature implementation processes. It was agreed that games from simulation and tycoon genres could be sources of reference for such improvements – although previous experiences suggested that Product Owners (POs) could be resistant to ideas inspired by video games.

4.1.2 Second meeting: scope indefinition and previous debates on game-like interfaces

At the second meeting between researcher and developers, it was discussed the possibility of changing SiVIEP's focus away from integrated simulation, due to difficult requirement eliciting and data acquisition. During a brainstorm for the redesigned graph and timeline functionalities, the development team presented a former interface redesign proposition, incidentally inspired by video games, which was discarded by POs. In that occasion, the lack of emphasis on 3D and inconsistencies between proposed iconography and oil industry standards were presented as justifications. It was suggested that POs should be interviewed by the researcher, for a clearer take on their perspective.

4.1.3 Third meeting: interview with POs

During a third meeting, the researcher performed an interview with the POs, who stressed the difficulty in obtaining data and use cases for the software. Integrated simulation for oil and gas was described as a recent field of development, with no established standards, in which case SiVIEP's development was, itself, research. In fact, the software is regarded, within the company's research center, as a 'blue sky' project – meaning it is not focused on immediate needs, but rather dedicated to long-term, future perspectives. It was suggested that SiVIEP's true value would come from allowing professionals to see, in 3D, information they had only seen in 2D before. POs expressed the intention of halting development of new VR features until desktop version was complete and disseminated, although acknowledging VR's potential for engaging users and hosting multidisciplinary meetings. Regarding software interface and immersion, they have expressed the desire for more intuitive interface, especially for 3D object placement and scene building. Navigation had been recently improved, but could be better, possibly allowing 3D navigation control from a 2D mini-map, which should be intuitive for engineering professionals. Since the VR experience was controlled via Nintendo's Wiimote controller without a sensor bar, pointer functionality was unavailable. Therefore, contextual

menus, radial if possible, were preferable – in which case menu options would be selected through directional pad. While one of the POs was well versed in games, particularly ones of simulation and tycoon genres, the other was not an active player nor very familiar with the media – although very informed about its potential for education and science, and willing to learn more about the subject.

4.1.4 End of Stage 1

By the end of Stage 1, production of the first low-fidelity prototype had initiated. Main briefed requirements gathered from the initial meetings were: (a) Redesign of *Graph* and *Timeline* features for unified Desktop and VR experiences; (b) Control schemes that are clear and intuitive for both game playing and non-game playing audiences.

4.2 Stage 2: low-fidelity prototypes production and reception

Research for the low-fidelity prototype was based around a number of different concepts. First, the use of Wiimote controller and Nintendo's tradition of developing for casual audiences led us to look at their products (specially Wii-based ones). Second, due to the proximity of SiVIEP's intended use with simulation games, we have investigated titles such as *Sim City* and *Cities in Motion* for an understanding on mini-maps and scene navigation. Third, aiming at higher levels of familiarity for non-playing audiences, we have looked at Smart TV interfaces. Although diverse in nature, those concepts were combined in a prototype in which a panel, equipped with a side-bar menu, contained a *graph* window from which users could navigate the virtual environment. On-screen instructions were displayed for clarity (Figure 4). The prototype also included the *Timeline* feature. Low-fidelity prototypes were made in *Adobe Illustrator* on top of screenshots taken from the actual software, and presented on computer monitors.
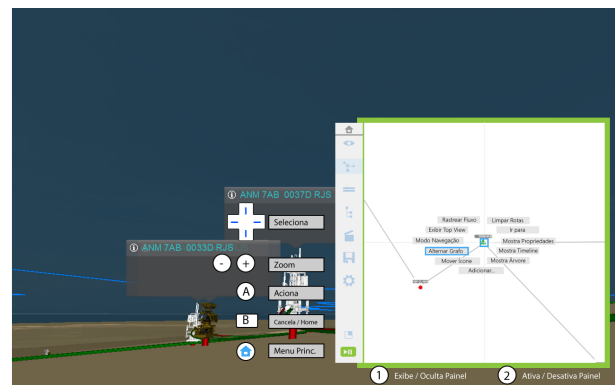

Figure 4: First iteration of the low-fidelity prototype.

4.2.1 Fourth meeting: low-fidelity prototype

The fourth meeting featured a demonstration of the prototype to a member of the development team, who made a series of observations to be considered for the next version of the prototype. The second version would have the side-bar menu removed, and the control system reworked to resemble Smart TV interface even more.

4.2.2 Research for new low-fidelity prototype

For the second iteration of the prototype, besides addressing points raised during a previous deliberation, new usability aspects were researched. More pronouncedly, a quick access to time navigation was devised, making it easier to be accessed during spatial navigation. Interestingly, such feature was inspired by the control scheme

of platform game *Braid*, in which the player can pause, rewind, resume and fast forward the development of in-game actions. This new feature was not designed to replace the *Timeline* (Figure 5) – which includes many more capabilities than playback control – but rather complement it.
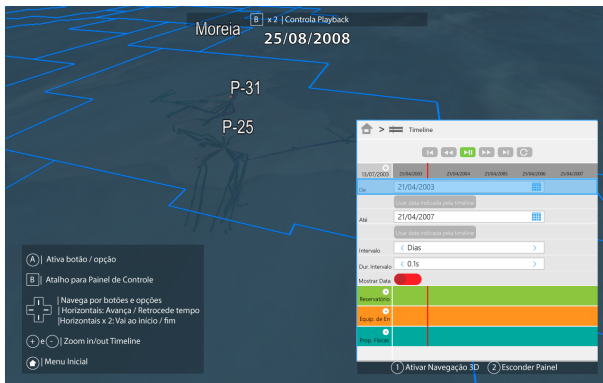

Figure 5: Timeline window and playback control instructions.

4.2.3   Fifth meeting: new prototype reception and changes in scope

During the fifth meeting, a new version of the prototype was presented. POs expressed their intention to reposition SiVIEP as a sandbox construction tool where to experiment new configurations for oil and gas enterprises. That change in focus would require emphasis on scene building tools, including capabilities for easily placing objects in the scene, drawing connections between objects, creating and loading terrain, and other similar features. Regarding the prototype itself, most positive comments were dedicated to the new playback control scheme. It was suggested an extended use of radial menus for functionality access, as opposed to its application on the *Main* and *Graph* panels only.

4.2.4   End of Stage 2

By the end of Stage 2, it was agreed that new requirements for SiVIEP as a construction tool would be addressed through further game design research and, this time, presented as a high-fidelity prototype.

4.3   Stage 3: further design research and high-fidelity prototype

Forced to refocus, participants would be, from now on, concerned about a new set of functionalities – and ways of presenting them.

4.3.1   Sixth meeting: refocusing

The sixth meeting involved developers, designer, and researcher. The meeting was dedicated to (a) reviewing research development and methods (b), presenting and generating ideas for SiVIEP's interface, and (c) establishing requirements and scope for the next prototype.

Regarding research review, besides summarizing the research so far, the idea of looking at games for solutions in interactivity and usability was reinforced. Screenshots for examined video game titles (*Cities in Motion 2*, *Democracy 3*, *Mini Metro*, *Another Code*, *Sim City Creator*, and *Braid*) were displayed along descriptions of features they have inspired (Figure 6).

Ideas for new functionalities were considered. It was discussed the possibility of including a tool for drawing pipelines between objects, possibly featuring a counter for distance, length and required construction material. Although promising, such functionality was

Figure 6: Cities in Motion 2 (top) and Braid (bottom): References for Graph navigation and playback control, respectively.

left to be possibly incorporated in a later stage of product development.

On the subject of prototype scope, it was established that its goal would be allowing users to put together a scene similar to the pre-configured scene used in SiVIEP demonstrations. It should allow users to load 3D models into predefined spatial locations – but also illustrate how they could be manually placed. It was suggested that an interactive, high-fidelity prototype would be a better option for contextualizing new features and demonstrating usability improvements. Finally, it was decided that the prototype would make an extensive use of radial menus – an interaction model for which examples could be found in both casual and hardcore games (Figure 7) and that should be, therefore, suitable for users of varied computer literacy levels.

Figure 7: Viva Piñata (left) and Mass Effect (right): Examples of radial menus in casual and hardcore games, respectively.

4.3.2   Seventh meeting: refining scope

The seventh meeting, featuring the same participants as the previous one, was held in order to refine prototype scope and require-

ments. It was decided that users should select, from an initial screen, which geographic region would be loaded in the scene. References for such interface included *Sim City 4* game selection menu and, also, a redesign concept proposed by a SiVIEP's designer over a year before (Figure 8).
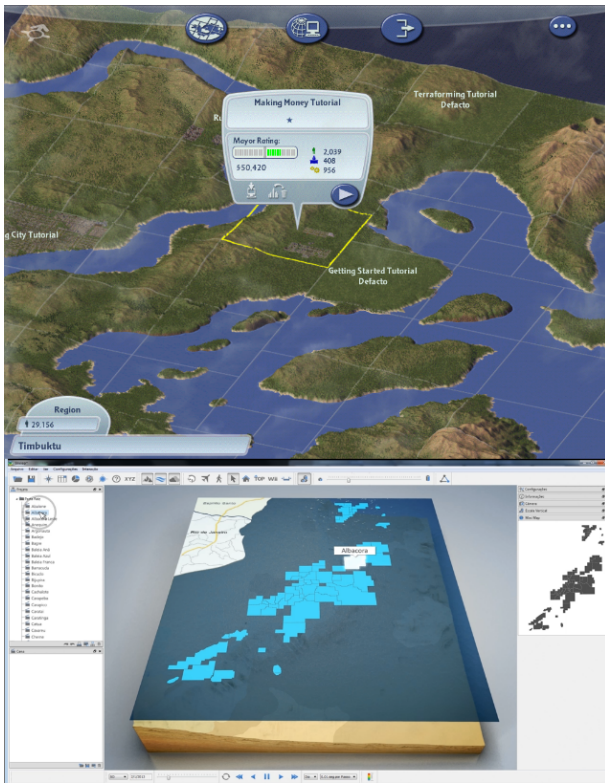

Figure 8: Sim City 4 (top) and SiVIEP redesign proposal from 2014 (bottom): References for scene selection.

4.3.3 High-fidelity prototype production

It was decided that the next prototype would be high-fidelity and dynamic, and produced with the help of *Unity* game engine and SloubiTools' *Circular Menu* plugin. Prototype would include initial stages of work intended in SiVIEP: (1) initial screen for selection of geographic limits of new virtual scene; (2) virtual environment including terrain of chosen geographic location; (3) activation of dialog screen for importing objects via radial menu; (4) use of *import* dialog screen; (5) display of imported objects in virtual space. Furthermore, functionalities would be designed for use in both desktop and VR environments.

The prototype's initial screen, which can be seen in Figure 9, featured a dialog box for confirming user's selection.

Upon user's confirmation, the initial terrain would be replaced by another one, representing the selected geographic location. From that scene, it was possible to activate a radial menu for command access (Figure 10). The only working commands for the prototype were the ones associated with object placement.

During prototype production, it was discussed that there should be different radial menu options for importing objects depending on whether they included embedded geolocation data or not. In the first case, the radial menu option would activate a dialog window for object selection that would include all kinds of objects (equipment, structures, reservoirs, etc.) (Figure 11). In the second case, another radial menu would be revealed (Figure 12), from which it would
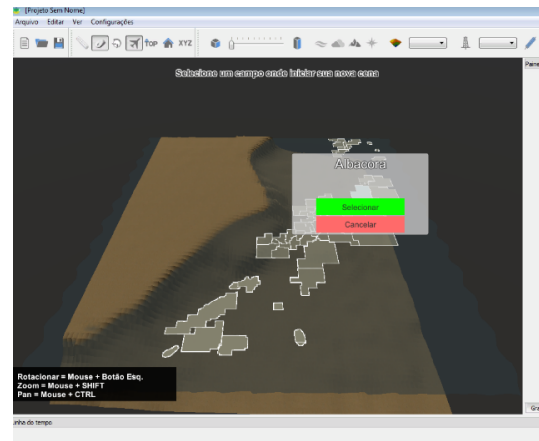

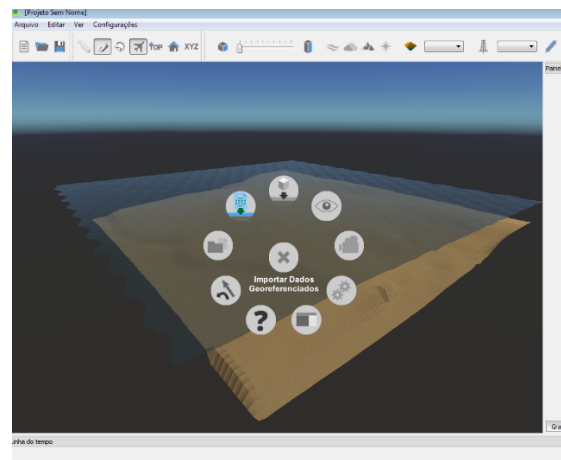Figure 9: High-fidelity prototype initial screen.

Figure 10: Activated radial menu.

be possible to open specific dialog windows for specific types of objects.
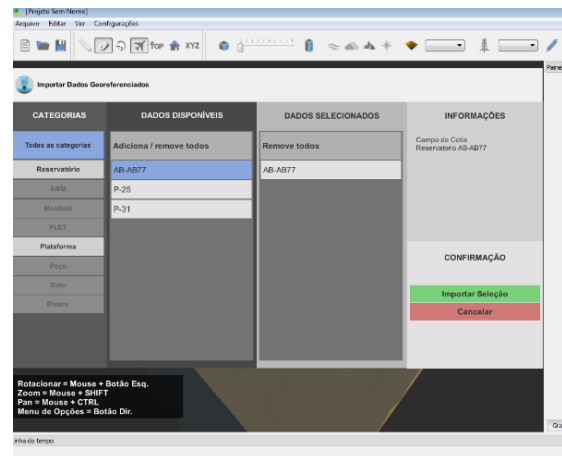

Figure 11: Dialog screen for geolocation-enabled objects.

Regarding the use of video games as reference, it must be added that dialog windows for object selection were influenced by the
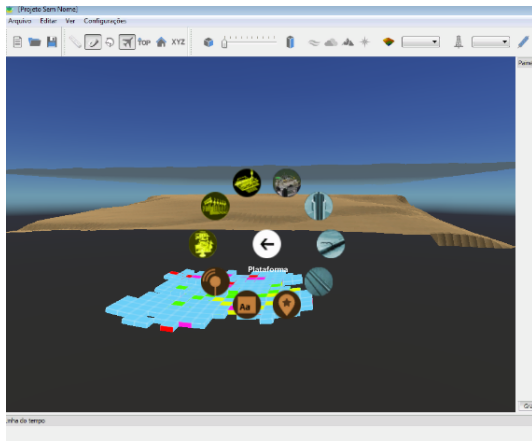
Figure 12: Radial menu for loading specific type objects. In the background, a reservoir imported via geolocation-enabled object dialog screen.

weapon selection menu found in *The Elder Scrolls V: Skyrim* (Figure 13).

Figure 13: The Elder Scrolls V: Skyrim - Weapon selection menu.

Once all planned features were incorporated into the prototype, it was decided that the functionality for positioning objects in virtual space should be implemented as well. That feature was particularly challenging to design, as it demanded: (a) intuitive control for desktop and VR modes, and (b) high precision for latitude and longitude input (up to seven digits each). Once again, simulation and tycoon games were used as reference: games such as *Sim City Build It* and *Farmville* (Figure 14) feature positioning tools that work with many input methods (mouse and keyboard, directional pads, and even touchscreens), while being effective and appealing to casual audiences.

Figure 14: Sim City Build It (left) and Farmville (right): references for positioning tool functionality.

Despite their ease-of-use, researched positioning tools lacked the

necessary precision required by scientific software in general, and by SiVIEP in particular. To allow for accurate positioning, two solutions were designed: (1) Numerical inputs for Latitude and Longitude were included. (2) When using arrow keys or directional pads, the increment to the object's position would be inversely proportional to the closeness between that object and the user's point of view (Figure 15). In this case, zooming into the object would lead to smaller increments, allowing for accurate positioning. Interestingly, this functionality was inspired by another SiVIEP feature: a multiscale navigation system that allows camera speed to be adjusted depending on whether there are complex objects nearby [43].
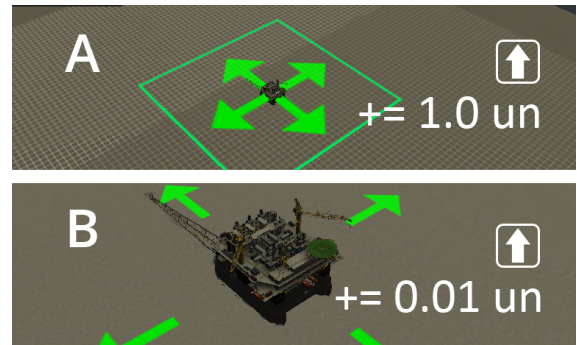

Figure 15: Pressing a directional key: in case 'A', the platform's position would increase in 1.0 unit; in case 'B', in only 0.01 unit.

In addition to the prototyping of selected features, all existing SiVIEP's functionalities (and also new ones) were mapped for eventual implementation in the radial menu system (Figure 16).
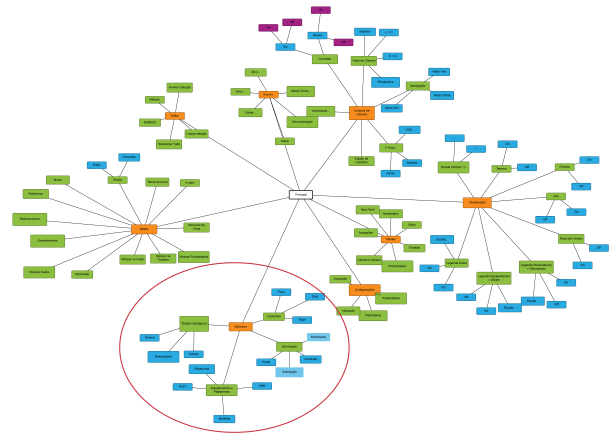

Figure 16: Totality of the mindmap illustrating SiVIEP's functionalities as radial menu options. Prototyped features are within the red circle.

4.4 Prototype reception and investigation closure

The high-fidelity prototype was presented to the development team during a brief informal meeting – the eighth one. Reception to the proposed interface was positive, and it was decided the prototype should be presented to POs. Four weeks later, during the ninth meeting, a presentation for the POs was arranged. Although well received by developers and POs, ideas presented in the prototype failed to be considered for implementation for a basic reason: the development of a radial menu system with actual software production tools and frameworks in use would demand an investment of

time and personnel that was seen as excessive – especially considering combined desktop and VR system requirements. Formal investigation was, then, brought to a halt. All parts remain, to this day, open for debate and advice on gamification and usability. Scene building tools were implemented months later, for desktop PC only, as a *wizard* tool organized into sequential steps – a feature more easily supported by SiVIEP's User Interface framework.

The exploration of video game-inspired interaction design was considered, overall, positive, since it has provided a gain of knowledge on the potential of design elements from games applied to scientific software. Perhaps more importantly, it has raised awareness to issues regarding the transition between design phase and implementation, and also issues regarding the design process itself. This knowledge, a positive outcome of this investigation, is more extensively discussed throughout the next section.

5 LESSONS LEARNED

Despite the failure to implement gamified functionalitites into the actual software, the research process allowed us to experiment, investigate, and gain knowledge on many relevant aspects of scientific software gamification design.

5.1 On scientific software particular challenges

Scientific software development often presents specific challenges. In our case, some of them had considerable impact on design and development, and often guided the research process.

5.1.1 Difficulties in requirements elicitation

The difficulty in obtaining requirements is regarded as a frequent obstacle in scientific software development. In this field of work, requirements have been described as "volatile"[36] and "emergent" [37]. In our particular case, changes in software scope demanded, at a particular stage, the abandonment of a redesigned group of functionalities, in order to focus on a new set of priorities. This difficulty suggests that the design process should be flexible enough to afford such changes in scope and focus, but, at the same time, be cohesive enough so that initially discarded ideas can be revisited in the future. Likewise, the design itself should make use of mechanics and design elements that support flexibility. In our case, the radial menu system was seen as an adequate option – not just for its usability, but also for potentially allowing the inclusion, exclusion and substitution of features in a non-intrusive manner. In that sense, a similar observation on the use of pop-up menus in cardiac bioelectricity visualization was made by MacLeod et al. [25].

5.1.2 Difficulties in validation

Scientific software output can be difficult to validate, as it has to be compared to real-world data, calculations done by hand, benchmarks based on other models, and/or judgement from field specialists [36]. In SiVIEP's case, it was not clear how to establish predictive models and detect relationships between data obtained from different sources. This suggests that the "machine-based arbitration" [11, p.21] present in video games, which judges players' performance, might be challenging to apply to scientific simulation, as this arbitration relies on comparing results obtained by the user to ones recognised as optimal by the system.

5.1.3 Attention to accuracy, precision and data quality

Correctness and precision are regarded as two of the most important qualities of scientific software [14]. Video games, on the other hand, can afford to be as simplified as they need to, in order to preserve gameplay. A clear example of this contrast could be seen during the prototyping of the positioning tool described at the end of subsection 4.3 and illustrated in Figure 15. Originally based on similar functionalities present in titles such as *Sim City* and *Farmville*, the arrow-based control scheme went through two modifications so it could achieve the necessary degree of precision. First, the inclusion of field texts for typing in numeric values. Second, the design of an additional mechanic, inspired by multiscale navigation, for increasing input accuracy. This need for accuracy suggests that video game-based control schemes might need to be adapted to satisfy scientific software needs for precision – either through its combination with more conventional solutions (such as the keyboard input) or through creative modification (multiscale positioning tool).

5.1.4 Iterative, incremental changes in complex systems

Emergent requirements and flexible scope demand a culture of iterative, incremental changes [36],[1],[47]. Accordingly, users often prefer small and incremental changes in the way they work [7]. Moreover, scientific software lifecycle can be measured in decades [4],[36],[18], grow beyond initial estimatives [4], and make use of more traditional, low-level programming languages – more efficient and more reliable, but more difficult to implement media-rich, highly interactive features. Underestimating these principles was a fundamental reason for not making the transition from prototype to actual implementation. Overall, regardless of improved usability, designed features would demand too much time and human resources to implement. This suggests that, especially when designing for already functioning software, one should adopt a conservative mindset regarding opportunities for implementation.

5.1.5 User-base demographics

Younger users have been reported to be more receptive to gamified work environments than older ones [30] The same could be said about the acquisition of skills in digital environments [46]. Although younger scientists can be enthusiastic about video game-inspired software[15], senior professionals migh offer resistance to this approach. In our case, since the software in question is, in fact, directed towards professionals in managerial levels and, therefore, more likely to be included in older demographics, our research had to contemplate strategies to address that audience. In that sense, the presence of one of the POs – who does not identifies himself as a "gamer", is older than the avarage gamer, and is, himself, in managerial level – was very helpful, as it provided a counterpoint to other team members' perspective (all identified as gamers), and serving as an example of the software's typical target-base.

5.1.6 Generation of insight

More than generating data, scientific software must provide insight [13]. The difficulty of obtaining insight into scientific problems using software tools is, currently, an issue[3]. In our case, the challenge was in improving current data visualization methods from SiVIEP, and also provide new ways of visualizing data within a 3D interative simulation. Two separate solutions were proposed: First, inspired by *Cities in Motion 2* extensive use of data visualization, it was proposed that SiVIEP's data graphs should represent temporal series rather than values from a single moment in time (Figure 17).

Figure 17: SiVIEP's current data panel (left) and Cities in Motion2 (right).

The second suggestion regarding the facilitation of insight was the time navigation feature described in subsection 4.3. Later, a team member observed that this function was not only present in games, and was probably originated from more traditional video playback control present in platforms such as AppleTv. Still, it was *Braid*'s remarkable combination of temporal and spatial navigation that made it adequate to SiVIEP's purposes.

5.2 On gamification research and design

The research process was very useful for identifying ways gamification could be approached, explored and designed.

5.2.1 Approaches to gamification

Gamification can be described as the act of restructuring an activity as a game, in which case gamified initiatives typically include goal-oriented game design elements such as points, scoreboards, winning conditions, competition, etc. This approach is often refered to as *gameful* design [10]. On the other hand, it can also convey the transposition of non-structural elements – such as interaction design patterns, visually pleasing aesthetics, etc. – from games to other media. This second approach was privileged in our research, as user's interaction with the system was the main issue at hand. Another reason for not structuring the software activity as a game is connected to aforementioned difficulties in scientific software: requirement eliciting and validation. In our case, it was not clear yet how existing data would be effectively integrated into a simulation in ways that preferable outcomes could be established and user performance, measured. In that sense, the only suggestion was made by SiVIEP's designer, who proposed a pipeline drawing tool featuring length, material, and cost calculator. Moreover, it was interesting noticing that *Sim City*, a major influence on the gamification process, was itself created as an experimentation on urban simulation, and referred to as a software toy, as opposed to a game, by its creator Will Wright [35, p. 439]. This suggests that this kind of free, sandbox mode play can be more appropriate for activities where creative experimentation is desired.

5.2.2 Researching design elements

Inspiration for gamified functionalities were mostly taken from existing game features. These were recalled during brainstorm sessions, or identified through research of specific titles that met any of the following criteria:

(a) Thematic proximity. In our case, tycoon and simulation games were popular choices, as they usually simulate enterprise building and management. This suggests that game features and mechanics would be adequate to the software.

(b) Intended functionality. Game titles with no thematic proximity can still suggest interesting functionalities to the software - in our case, an example would be our appropriation of *Braid*'s playback control.

(c) Input and Output devices. The use of special equipment – such as the *Wiimote* in our case – can demand an investigation of titles made specifically to support it.

It is also important to acknowledge the existence of design elements which are common to many games and, therefore, considered established conventions (e.g.: rectangular building placement tools, featuring directional arrows, present in both *Farmville* and *Sim City Built It*).

5.2.3 High-fidelity prototyping

The development of a high-fidelity prototype was substantial part of the research process. A reason for that was the perceived need for demonstrating, in a tangible way, how functionalities would translate from games into the application. In that sense, high-fidelity prototypes supported a clear **contextualization** of new features by bringing them into the software's familiar interface. Also, interactivity allowed an increased **understanding of necessary user skills**, since users could actually operate the proposed system's functionality. Furthermore, high-level prototyping allowed for better **polish** – in which case the user's experience was supported by aesthetically pleasant elements such as sofisticated 3D graphics that would be absent from low-fidelity prototypes. Polish, as proposed by Swink, can be a crucial element in player's experience of interactive virtual environments [42]. In our particular case, it is important to highlight that SiVIEP makes use of high-quality, visually appealling, 3D models and textures, produced with similar tools and techniques to the ones used in video games, as opposed to ultra-detailed, but visually flat, 3D models normally associated with engineering software. In fact, 3D models and textures used in the high-fidelity prototype were the actual ones from SiVIEP.

On the other hand, high-fidelity prototyping can lead to unfeasible propositions. In our experience, the game engine capability of supporting sophisticated, highly interactive user interfaces have somewhat influenced design decisions towards a complex model that could not be incorporated into the actual software. For that reason, we propose that prototypes developed through similar tools have their functionalities restricted to techniques that have been previously tried and tested by the development team in the actual development environment – or at least to ones that can be tested promptly. In that sense, a clear, realistic sense of system requirements and limitations is essential.

5.3 On the underlying research

Within the context of its underlying doctoral research, this study has raised awareness to methodological issues that should be addressed in subsequent studies. More pronouncedly, it has become apparent that the use of action-research methodology could be adjusted and, possibly, combined with different methods and approaches.

5.3.1 Action-research methods and outcomes

By structuring this study as an action-research, we have employed methods and dynamics associated with that methodology, such as meetings and interviews. However, the process seemed to be more beneficial to the researcher – who could compare results to theoretical background and literature review – than to other participants, who have arguably acquired knowledge on gamification and explored new ideas for their product, but have not consolidated that knowledge as implemented features. Moreover, the study failed to result in changes to the development process. This suggests that subsequent studies should be more attentive to methods for consolidating dynamics that can be incorporated in production.

5.3.2 Participatory design

Somewhat related to the point previously made, the difficulty in establishing a set of practices during this action-research stands as an evidence of the need for developing and adopting practices and dynamics capable of fostering collaboration between researcher and other participants, possibly by increasing the physical presence of the researcher in development work space and use of conversational support such as message boards, online forums, etc..

5.3.3 Practice-based research

Throughout the study, the researcher himself has engaged in producing low and high-fidelity prototypes. To a certain extent, this experience has, directly or indirectly, led to some of the findings expressed in the Lessons Learned section. Should this approach be further incorporated, practice-based research methods should be thoroughly investigated and formally included into the underlying research.

6 CONCLUSION

Gamification, successfully applied to correlated areas such as citizen science and science education, could also help improving scientific software usability – which is an often overlooked aspect of that type of software. However, the use of video games as inspiration demands attention to idiosyncrasies and particularities of scientific software development and use. To better understand implications, we have conducted an action-research around the redesign of functionalities from an oil and gas software application. Although limited to a single case, the investigation could attest the impact of scientific software characteristics on gamification design. Attention to user demographics, correctness and insight have led to interesting user interface solutions inspired by design elements from video games – some of which had to be reimagined for scientific software needs. On the other hand, problems in validation, difficulties in requirements elicitation, and software complexity exerted influence on design decisions – such as not restructuring the software activity as a game, but rather redesigning its features to be more similar to ones found in video games . This approach also seemed as adequate to the exploratory, creative nature of the software's activity. Search for inspirational features was carried out through the investigation of games depicting similar activities (e.g.: tycoon/building simulation); featuring intended functionalities (e.g.: time manipulation); or supporting similar devices (e.g. Wiimote controller). Redesigned features were presented in low and high-fidelity prototypes. The latter, built with a game engine, seemed more effective in demonstrating proposed system functionalities, with a caveat: the ease in prototyping such features was not compatible with the difficulty and cost of actually implementing them – which were regarded as impediments, despite perceived improvement in functionality. In this case, design phase should be clearly informed of limitations in development framework and resources, and also aim at small, iterative changes, otherwise risking implementation. Further research will focus on ways of mitigating that risk by improving communication and collaboration between all participants, and also by investigating complimentary design and research methods. Additionally, it might approach gamification from a goal-oriented, gameful perspective. Moreover, subsequent similar studies could be conducted with different software development teams, as the underlying doctoral research, upon which this study was based, moves forward.

7 ACKNOWLEDGEMENTS

REFERENCES

[1] K. S. Ackroyd, S. H. Kinder, G. R. Mant, M. C. Miller, C. A. Ramsdale, and P. C. Stephenson. Scientific software development at a research facility. *IEEE software*, (4):44–51, 2008.

[2] Z. Ahmed, S. Zeeshan, and T. Dandekar. Developing sustainable software solutions for bioinformatics by the butterfly paradigm. *F1000Research*, 3, 2014.

[3] C. R. Aragon, S. Poon, and C. T. Silva. The changing face of digital science: new practices in scientific collaborations. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 4819–4822. ACM, 2009.

[4] V. R. Basili, J. C. Carver, D. Cruzes, L. M. Hochstein, J. K. Hollingsworth, F. Shull, and M. V. Zelkowitz. Understanding the high-performance-computing community: A software engineer's perspective. *IEEE software*, 25(4):29, 2008.

[5] S. Boeykens. Using 3d design software, bim and game engines for architectural historical reconstruction. *CAAD Futures, Liege, Belgium*, 2011.

[6] P. De Matos, J. A. Cham, H. Cao, R. Alcántara, F. Rowland, R. Lopez, and C. Steinbeck. The enzyme portal: a case study in applying user-centred design methods in bioinformatics. *BMC bioinformatics*, 14(1):1, 2013.

[7] D. De Roure and C. Goble. Software design for empowering scientists. *Software, IEEE*, 26(1):88–95, 2009.

[8] P. Deemer, G. Benefield, C. Larman, and B. Vodde. The scrum primer. *Scrum Primer is an in-depth introduction to the theory and practice of Scrum, albeit primarily from a software development perspective, available at: http://assets. scrumtraininginstitute. com/downloads/1/scrumprimer121. pdf*, 1285931497:15, 2010.

[9] S. Deterding. The lens of intrinsic skill atoms: A method for gameful design. *Human–Computer Interaction*, 30(3-4):294–335, 2015.

[10] S. Deterding, D. Dixon, R. Khaled, and L. Nacke. From game design elements to gamefulness: defining gamification. In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pages 9–15. ACM, 2011.

[11] J. Ferrara. *Playful Design*. Rosenfeld Media, 2012.

[12] J. Hamari, J. Koivisto, and H. Sarsa. Does gamification work?–a literature review of empirical studies on gamification. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pages 3025–3034. IEEE, 2014.

[13] R. Hamming. *Numerical methods for scientists and engineers*. Courier Corporation, 2012.

[14] L. Hatton and A. Roberts. How accurate is scientific software? *Software Engineering, IEEE Transactions on*, 20(10):785–797, 1994.

[15] H. Javahery, A. Seffah, and T. Radhakrishnan. Beyond power: making bioinformatics tools user-centered. *Communications of the ACM*, 47(11):58–63, 2004.

[16] D. F. Keefe. Integrating visualization and interaction research to improve scientific workflows. *Computer Graphics and Applications, IEEE*, 30(2):8–13, 2010.

[17] G. Keenaghan and I. Horvath. State of the art of using virtual reality technologies in built environment education. In *TMCE 2014: Proceedings of the 10th International Symposium on Tools and Methods of Competitive Engineering, Budapest, Hungary, 19-23 May 2014*. Delft University of Technology, 2014.

[18] D. Kelly. Determining factors that affect long-term evolution in scientific application software. *Journal of Systems and Software*, 82(5):851–861, 2009.

[19] D. Kelly. An analysis of process characteristics for developing scientific software. *Innovative Strategies and Approaches for End-User Computing Advancements*, page 347, 2012.

[20] Z. Kosmadoudi, T. Lim, J. Ritchie, S. Louchart, Y. Liu, and R. Sung. Engineering design using game-enhanced cad: The potential to augment the user experience with game elements. *Computer-Aided Design*, 45(3):777–795, 2013.

[21] W. Li, T. Grossman, and G. Fitzmaurice. Gamicad: a gamified tutorial system for first time autocad users. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 103–112. ACM, 2012.

[22] T. T. Lo, M. A. Schnabel, S. Aydin, and K. Shi. Using gamification for collaborative mass-housing design process.

[23] S. Loynton, D. Sloan, J.-M. Burel, and C. Macaulay. Towards a project community approach to academic scientific software development. In *E-Science Workshops, 2009 5th IEEE International Conference on*, pages 120–124. IEEE, 2009.

[24] C. Macaulay, D. Sloan, X. Jiang, P. Forbes, S. Loynton, J. R. Swedlow, and P. Gregor. Usability and user-centered design in scientific software development. *IEEE Software*, 26(1):96, 2009.

[25] R. S. MacLeod, C. R. Johnson, and M. A. Matheson. Visualization of cardiac bioelectricity: a case study. In *Proceedings of the 3rd conference on Visualization'92*, pages 411–418. IEEE Computer Society Press, 1992.

[26] M. E. A. Marasco, L. Behjat, and W. Rosehart. Integration of gamification and creativity in engineering design.

[27] J. Moloney. Videogame technology re-purposed: Towards interdisciplinary design environments for engineering and architecture. *Procedia Technology*, 20:212–218, 2015.

[28] B. Morris, S. Croker, C. Zimmerman, D. Gill, and C. Romig. Gaming science: the gamification of scientific thinking. *Frontiers in psychology*, 4:607, 2013.

[29] R. J. Murphy. *Facilitating Citizen Science Through Gamification*. PhD thesis, Memorial University of Newfoundland, 2015.

[30] F. Oprescu, C. Jones, and M. Katsikitis. I play at workten principles for transforming work processes through gamification. *Frontiers in psychology*, 5, 2014.

[31] C. M. Pancake. improving the usability of numerical software through user-centered design. *Quality of Numerical Software: Assessment and Enhancement, ed. B. Ford and J. Rice, Chapman & Hall*, 1996.

[32] D. M. Popa. Design case: Gamification of erp a user centered design approach. *CM CHI*, 13, 2013.

[33] D. M. Popa. Industry design case: Introducing gamification persona tool. *CM CHI*, 13, 2013.

[34] A. Raposo, L. Reis, V. Rodrigues, R. Pinheiro, P. Elias, and R. Cherullo. A system for integrated visualization in oil exploration and production. *IEEE Computer Graphics and Applications*, 36(3):10–16, May 2016.

[35] R. Rouse. *Game design: theory & practice*. Wordware Pub, Plano, 2001.

[36] R. Sanders. The development and use of scientific software. 2008.

[37] J. Segal and C. Morris. Developing scientific software. *Software, IEEE*, 25(4):18–20, 2008.

[38] J. Segal and C. Morris. Developing software for a scientific community: some challenges and solutions. *Handbook of Research on Computational Science and Engineering: Theory and Practice*, pages 177–196, 2011.

[39] D. Sloan, C. Macaulay, P. Forbes, and S. Loynton. User research in a scientific software development project. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, pages 423–429. British Computer Society, 2009.

[40] E. R. Souza and E. Souto. Utilizacao de heuristicas de jogos para avaliacao de um aplicativo gamificado. In *Proceedings of SBGames 2015*, pages 666–673. SBC, 2015.

[41] R. R. Springmeyer. Applying observations of work activity in designing prototype data analysis tools. In *Visualization, 1993. Visualization'93, Proceedings., IEEE Conference on*, pages 228–235. IEEE, 1993.

[42] S. Swink. *Game feel: a game designer's guide to virtual sensation*. Morgan Kaufmann, 2009.

[43] H. Taunay, V. Rodrigues, R. Braga, P. Elias, L. Reis, and A. Raposo. A spatial partitioning heuristic for automatic adjustment of the 3d navigation speed in multiscale virtual environments. In *3D User Interfaces (3DUI), 2015 IEEE Symposium on*, pages 51–58. IEEE, 2015.

[44] A. K. Thomer, M. B. Twidale, J. Guo, and M. J. Yoder. Co-designing scientific software: Hackathons for participatory interface design. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 3219–3226. ACM, 2016.

[45] D. Tripp. Action research: a methodological introduction. *Educacao e pesquisa*, 31(3):443–466, 2005.

[46] J. Webster and J. J. Martocchio. Turning work into play: Implications for microcomputer software training. *Journal of Management*, 19(1):127–146, 1993.

[47] G. Wilson, D. Aruliah, C. T. Brown, N. P. C. Hong, M. Davis, R. T. Guy, S. H. Haddock, K. D. Huff, I. M. Mitchell, M. D. Plumbley, et al. Best practices for scientific computing. *PLoS Biol*, 12(1):e1001745, 2014.

[48] C. Wolff. The case for teaching tool science taking software engineering and software engineering education beyond the confinements of traditional software development contexts. In *Global Engineering Education Conference (EDUCON), 2015 IEEE*, pages 932–938. IEEE, 2015.