Adopting virtual reality as a medium for software development process education

Conference Paper · May 2018

DOI: 10.1145/3202710.3203161

Adopting Virtual Reality as a Medium for Software Development Process Education

Ulas Gulec
Department of Computer Engineering, Middle East Technical University
Ankara, Turkey
Department of Computer Engineering, Çankaya University
Ankara, Turkey
ulasgulec@cankaya.edu.tr

Murat Yilmaz
Department of Computer Engineering, Çankaya University
Ankara, Turkey
myilmaz@cankaya.edu.tr

Veysi Isler
Department of Computer Engineering, Hasan Kalyoncu University
Gaziantep, Turkey
Department of Computer Engineering, Middle East Technical University
Ankara, Turkey
veysi.isler@hku.edu.tr

Rory V. O'Connor
School of Computing, Dublin City University
Dublin, Ireland
rory.oconnor@dcu.ie

Paul Clarke
School of Computing, Dublin City University
Dublin, Ireland
paul.m.clarke@dcu.ie

ABSTRACT

Software development is a complex process of collaborative endeavour which requires hands-on experience starting from requirement analysis through to software testing and ultimately demands continuous maintenance so as to mitigate risks and uncertainty. Therefore, training experienced software practitioners is a challenging task. To address this gap, we propose an interactive virtual reality training environment for software practitioners to gain virtual experience based on the tasks of software development. The goal is to transport participants to a virtual software development organization where they experience simulated development process problems and conflicting situations, where they will interact virtually with distinctive personalities, roles and characters borrowed from real software development organizations. This PhD in progress paper investigates the literature and proposes a novel approach where participants can acquire important new process knowledge. Our preliminary observations suggest that a complementary VR-based training tool is likely to improve the experience of novice software developers and ultimately it has a great potential for training activities in software development organizations.

CCS CONCEPTS

• **Computing methodologies → Modeling and simulation**; • **Software and its engineering → Agile software development**;

KEYWORDS

Virtual experience in software development; software process; virtual reality

1 INTRODUCTION

Software is a set of readable and executable instructions that can run on computers [21]. However, even though software can be easily defined, development of software including several different phases is a very complicated process [28]. Each phase of software development plays a significant role to produce profitable products [18]. Although it is an essential issue, most of the software projects cannot be successfully completed [7]. For this reason, the software development process should be fully educated to software practitioners in order to avoid undesirable situations occurring in the development of the software products. To accomplish this aim, it is necessary to provide an environment for the members in the software development teams where they can face the real life problems.

Virtual reality (VR) is a technology that provides an efficient and useful training environment for individuals [36]. This technology immerses the participants in a virtual environment designed to be similar to real life [20]. The aim is to experience the problems that people may face in real life in a virtual environment. Thus, individuals make fewer mistakes in real life, because, they gain experience against the problems they may encounter in real life via using virtual environment [10]. With this feature, VR has become one of the most popular workspaces of the last 20 years [29]. Due to this popularity, the amount of investment made in this area has significantly increased [6, 14]. Along with the increased investments, the number of VR projects in many different areas including medical [42], sport [16] and education [24] has increased rapidly.

These studies show that VR is an effective training tool that can be used to increase individuals' level of knowledge and experience in different working spaces. Hence, the aim of this study is to create a background of our thesis which aims to design a 3D interactive environment where the participants can experience the virtual software development process. Due to this environment, the effects of software practitioners can be further improved by increasing the individuals' level of experience about problems occurring in the software development process.

1.1 Training challenges for software development

There are a number of studies [4, 25, 40] in the literature showing that the software development methodologies are generally hard to be trained for software development teams. Especially, the agile development needs to be combined with other techniques such as "*user centered design*" [15] or "*model driven development*" [26] in order to be successfully deal with development challenges. The reason for these challenges is that the software development methodologies are hard to understand without real-life practices. Although software development methods are well-understood by experienced managers, they cannot easily transfer this tacit knowledge since novice practitioners do not fully comprehend the technical details of software development [9]. Therefore, members in a software development team have to play different roles during the software development project since they try to close each other's deficiencies to bring out a successful software product. For this reason, roles in software team members become exchangeable since the ultimate goal is to deploy the software product [23]. However, if an introvert person is involved in the project, this situation directly affects the quality of the software development process since the communication between the team members cannot be performed properly [39]. In particular, in some software projects, team members should self organize to successfully complete the software projects when their roles are not well-defined [38]. However, this situation can also directly affect the success of the project negatively, due to the fact that the workload within the team is not equal and that the self-organized people may not share information.

For the agile development, another training challenge is that a gap in communication between team members is usually filled with documentation of a project although agile development aims to provide quick deliver of the end product to the customer [27], however, this situation might cause a problem of onboarding of novice participants to a software project [34]. The novice member of a software team cannot follow onboarding steps of the project because of a lack of previous training about the project. In addition to the adaptation problem of the new project members, this case also affects the quality of the end product [32], because, if the number of new members in the project development team increases, the project cannot be developed properly since almost all of them have problems of adaptation. The last challenge is that the management of projects developed using agile development features is difficult because of the constant updating of project dynamics, time constraints and lack of a clear project plan [39]. The main reason for this difficulty is that the agile model provides customer-oriented project development process. The customers might be included into the project as a team member. This situation leads to continuous project updates since the demands of the customers are continuously embedded in the project. Most importantly, the complexity of the project increases when the number of requests of the customers causes managerial interventions. To deal with such issues, a novice software practitioner should have to be trained with different scenarios.

When the above paragraph is summarized to illustrate the training drawbacks of software development, the improvement of the following points are important to increase the quality of the software projects: *1- To increase the knowledge levels of the project members regarding the applied process*, *2- To better identify the roles of individuals in the project team*, *3- To give more detailed information team members about their duties in the project*, *4- To accelerate the adaptation process of new participants to the project* and *5- To choose more suitable people for the project*.

In order to fill these gaps, this study provides a virtual environment in which project members can develop themselves on the basis of the agile development since VR is one of the commonly used tool in the field of education. Next section will explain the benefits of VR applications when they are used to improve individuals' experience and knowledge levels.

1.2 Importance of Using Virtual Reality Applications in Hands-on Training

A literature review was conducted to find studies which have been completed for training a participant by developing a VR application. As a result of this survey, a number of studies was found in the literature investigating similar problem in different domains.

Fang and Teizer [17] developed a virtual environment to train both crane operators and ground personnel. The aim of this study is to increase the collaboration between crane operators and ground personnel by reducing the mistakes during a construction. This environment has been tested by construction graduate students. According to the results of this study, the practice in the virtual environment improves the skills of the participants. In another study by Bliss et al. [3], a virtual environment was designed to increase the navigation capabilities of firemen. In order to understand the effect of the virtual environment on the education of the individuals, participants divided into three different groups as blueprint, VR and no training. According to the results, the VR is an effective tool that can be used in education domain. Seymour et al. [37] has conducted a study to illustrate the effect of VR when it is used as a training tool. The aim of this study is to improve the abilities of the medical students by creating a virtual environment. A group of student has been equally divided into two different groups, VR and non-VR. According to the results obtained from this study, students who studied with VR have developed themselves more than the students who worked with traditional methods. In the same vein, Kandalaft et al. [22] has a study to improve the social abilities, attention and functioning of young adults who are diagnosed with autism at high levels. In Second Life, an area including offices, buildings, shopping shops, cafes, restaurants, schools and parks has been reserved for only this training program. This virtual environment has been used by eight participants. The results of this study illustrate that virtual environments can be used as an important educational tools to train people and enhance their abilities.

These studies in the literature show that VR is an appropriate platform used in the training of individuals in different working areas.

The aim of this paper is to explore the potential use of interactive approaches for software developer's hands-on training and propose VR approach based on situations and events originally taken from software development organizations.

2 RELATED WORK

There are several different studies in the literature that develop tools for training individuals about software engineering concepts. While most of these studies are 2D or board/card games, and involve many features of serious games, 3D virtual environments are also designed to enable individuals to experience a more active learning process in some other studies.

An example study in this area was completed by Baker et al. [2]. In this study, a card game, which supports traditional education methods, has been developed with the aim of increasing the level of experience of people about software development processes without involving real life risks. The game is multi-player and each player is given a task similar to the real life project. The players must perform these tasks as soon as possible to meet the customer's requests in accordance with their budget. According to the results of the study, this card game is a beneficial educational tool that can be used to support traditional education methods since it creates a competitive atmosphere among the players which encourages people to play.

In similar way, Bollin et al. [5] developed a simulation framework to experience different software development methodologies for the participants and to advise project managers or team leaders about decision-making processes by giving feedbacks based on their choices in the system. This framework is a computer application that contains elements similar to flowchart components. Hence, the participants can develop the projects or decide on a topic by connecting these elements to each other. In this way, the participants have a chance to improve their experience levels on project development without living real experiences.

The idea of using games in software engineering education is also supported by Hainey et al. [19] who produced a game that helps individuals about how the requirements of a project should be gathered and analyzed. In this game, the players should manage and complete the software projects with different roles such as team leader, system analyst, designer or project manager. While the players are managing the projects, they should pay attention the constrains which should be also determined by the players when they read the details of the project. The game was supported by non-player characters who direct the players about their tasks. This game was tested with 92 students. According to the results obtained from this study, this game is very useful to increase the knowledge levels of the participants about software engineering since it attracts the attention of people.

Similarly, Rusu et al. [35] invented an interactive game that teaches the students about the content of the maintenance phase of SDLC. The game dynamics of this newly developed game are very similar to game dynamics of classical tower-defence game. The bugs and errors in the system were represented as enemies and the projects were represented as towers. The players try to protect their towers from the enemies by using the available protection methods in the game. 18 students were selected to understand the efficiency of the game. The results of this study indicate that the games can be an alternative learning technique which can be effectively used in the education of software engineering topics.

When the 3D applications developed in this area are to be examined, Aydan et al. [1] designed a serious game to teach individuals the basics of ISO/IEC 12207:1995 in an enjoyable manner. For this game, a 3D virtual office environment supported by non-player characters was created to further impress the participants. This virtual environment was tested with 40 first-year students in the computer engineering department. The findings of this study illustrate that the level of knowledge about the ISO/IEC 12207:1995 of the participants has improved considerably.

Ye et al. [41] used Second Life as an education tool where the faculty members can have ability to answer the students' questions about software engineering via using the virtual environment. A virtual classroom similar to real classroom was designed in Second Life to provide communication between students and lecturers. This environment was tested with a group of students in software engineering department. The results indicate that a virtual teaching strategy is advantageous in enhancing the knowledge levels of students about software engineering field.

Another similar study was developed by Rodriguez et al. [33]. In this study, a 3D virtual seminar room was designed to teach software engineering students the practices of Scrum without having time and facility limitations. This virtual environment consists of virtual elements similar to real elements in a meeting room such as blackboard, charts and calendar. In this environment, the participants were allowed to move the virtual objects in order to increase the reality of the system. An example project was integrated into the system to demonstrate the flow of the Scrum methodology. In this way, when the project is developed with Scrum, the steps to be taken at the stage of the project are explained at the simulated Scrum meetings. The virtual environment was tested with 45 undergraduate students. According to the results of this study, this virtual environment is a beneficial tool that can be used to explain the flow of the Scrum method to the students.

Parsons and Stockdale [30] conducted a study in which a virtual world was designed by using Open Wonderland as a distance learning tool to educate the participants about the properties of agile software development methodology such as user stories, features in user stories and team collaboration. In this study, it is expected that the knowledge levels of participants about agile software process will be increased by simple development of a project that is not related to software engineering. The findings of this study clarify that although the developed tool requires significant improvements such as whole project development process and more realistic project scenario, it can be used for educating individuals about software engineering processes.

3 DISCUSSION

In light of these remarks, we claim that novice software practitioners could benefit from a training tool to gain some virtual experience. In such a VR-based simulation, trainees could take on

Table 1: Comparison Table with the Existing Studies in the Literature

Authors	Tool	High Reality	All Project Development	NPCs with AI	Different Stories
Baker et al. [2]	Card Game	✗	✓	✗	✓
Bollin et al. [5]	Desktop Application	✗	✓	✗	✓
Hainey et al. [19]	Digital 2D Game	✗	✗	✓	✗
Rusu et al. [35]	Digital 2D Game	✗	✗	✗	✗
Aydan et al. [1]	3D Application	✗	✗	✓	✗
Ye et al. [41]	3D VR Application	✓	✗	✗	✗
Rodriguez et al. [33]	3D VR Application	✗	✓	✗	✗
Parsons and Stockdale [30]	3D VR Application	✗	✗	✗	✗

the role of a new software developer being recruited into a virtual software development firm who should work alongside a set of virtual characters, all the time being guided through artificially intelligence means. Our ultimate aim is to a unique time-line where certain virtual characters serve as the company guide; some others might offer their long-gathered expertise and insight as participants navigate through a virtual software project based on real events and data. The required data for creating software development training scenarios are planned to be gathered from a middle size software company. In addition, our exclusive viewpoint should draw participants from a classical video game experience into a virtual world of software development itself. Through interacting, participants would acquire hands-on experience on practicing the software development process.

One of the challenges associated with software processes concerns the amount of process adaptation that takes place in individual software development settings [13]. Part of this challenge is related to the large number of different development approaches and tools available, with many studies reporting that process tailoring or adaptation is a common occurrence. We therefore find that software development process education is not so simple as to educate practitioners and prospective practitioners on the various different approaches but we must also find ways to assist them in continually matching a process with its situational context [12]. The type of multiple scenario generation possible in VR can be of assistance in building up experience of the various different challenges that can arise and can point trainees towards various different solutions to their challenge. In a sense, it can be helpful in rapidly acquiring experience of process evolution and adaptation without the need for many years of on the job training. Therefore, whether our proposal be best suited to this task or not, the potential impact of VR on this aspect of software process education can be considered to be relatively high.

4 CONCLUSIONS AND FUTURE WORK

According to the results obtained from the literature review, it is necessary to fill the missing points of software development in order to increase the quality and success of the software projects. Although there exist a number of applications that try to fix these missing points, there are still some drawbacks of these systems that should be completed. Table 1 shows these deficiencies on a study basis.

When these studies are analyzed in detail, the main weaknesses can be summarized as:

- **Low/Limited Increased Reality:** The most important feature that should be included in the designed tools which bring real life experience to individuals without the real-life risks is reality to the real environment. This is a very critical property since the success of the environment directly affects the learning level of the people in a positive manner [31]. Although it is an essential issue for training tools, the card games and 2D games cannot fully provide this feature since these platforms do not have enough hardware and software functions to detach the players from the real environment. Hence, this situation decreases the sense of presence which is the most significant factor that shows the success of the designed environment [11]. 3D environments can provide the sense of presence, however, the above studies did not give much importance to this issue since they did not measure their participants level of sense of presence.
- **Missing Whole Project Development Processes:** As it is known, the software is a product obtained after several different phases in SDLC. Although the development of a software is a long process, the existing studies aim to increase the participants' level of knowledge about the specific phases. None of them can inform the users about the whole project development time since code implementing or testing issues can be given as an example topics in SDLC that were not discussed in the above studies.
- **Limited Number of Stories:** The above studies only focused on one project as an example test project. In addition, some of these sample projects were not related with the software engineering and the other ones were not large scale projects like in real life. Therefore, this case can be shown as an important drawback of the existing systems, because, analyzing only one project limits the development of individuals since the requirements of real life projects are very different from each other. Hence, the participants cannot face to different problems occurring in the different types of the projects.
- **No Artificial Intelligent NPCs:** Continuous repetition of people in a topic increases the success of the people [8]. Although repetition is a critical factor that affects the personal evolution, the platforms designed in above studies require multiple players to experience the actions in the software development phases. If a person cannot find other players, s/he

will not use the system. This is the most significant drawback of the existing systems since they are not continuously available.

To sum up, this study purposes an interactive virtual environment to help participants gain experience based on the tasks of SDLC by enhancing the above drawbacks of the existing studies in the literature. Due to this environment, the participants have a chance to face development problems and conflicting situations with some of distinctive virtual personality characters borrowed from real software development organizations without real-life risks.

REFERENCES

[1] Ufuk Aydan, Murat Yilmaz, Paul M Clarke, and Rory V OŠConnor. 2017. Teaching ISO/IEC 12207 software lifecycle processes: A serious game approach. *Computer Standards & Interfaces* 54 (2017), 129–138.

[2] Alex Baker, Emily Oh Navarro, and Andre Van Der Hoek. 2005. An experimental card game for teaching software engineering processes. *Journal of Systems and Software* 75, 1 (2005), 3–16.

[3] James P Bliss, Philip D Tidwell, and Michael A Guest. 1997. The effectiveness of virtual reality for administering spatial navigation training to firefighters. *Presence: Teleoperators and Virtual Environments* 6, 1 (1997), 73–86.

[4] Stefan Blomkvist. 2005. Towards a model for bridging agile development and user-centered design. *Human-centered software engineeringŮintegrating usability in the software development lifecycle* (2005), 219–244.

[5] Andreas Bollin, Elke Hochmüller, and Roland T Mittermeir. 2011. Teaching software project management using simulations. In *Software Engineering Education and Training (CSEE&T), 2011 24th IEEE-CS Conference on*. IEEE, 81–90.

[6] NM Bouchlaghem and IG Liyanage. 1996. Virtual reality applications in the UK's construction industry. *CIB REPORT* (1996), 89–94.

[7] Eric J Braude and Michael E Bernstein. 2016. *Software engineering: modern approaches*. Waveland Press.

[8] Marcus Buckingham. 2005. What great managers do. *IEEE Engineering Management Review* 33, 2 (2005), 3–10.

[9] Martina Ceschi, Alberto Sillitti, Giancarlo Succi, and Stefano De Panfilis. 2005. Project management in plan-based and agile companies. *IEEE software* 22, 3 (2005), 21–27.

[10] Li-Keng Cheng, Ming-Hua Chieng, and Wei-Hua Chieng. 2014. Measuring virtual experience in a three-dimensional virtual reality interactive simulator environment: a structural equation modeling approach. *Virtual Reality* 18, 3 (2014), 173–188.

[11] Burak Çiflikli, Veysi İşler, and Uğur Güdükbay. 2010. Increasing the sense of presence in a simulation environment using image generators based on visual attention. *Presence: Teleoperators and Virtual Environments* 19, 6 (2010), 557–568.

[12] Paul Clarke and Rory V OŠConnor. 2012. The situational factors that affect the software development process: Towards a comprehensive reference framework. *Information and Software Technology* 54, 5 (2012), 433–447.

[13] Paul Clarke, Rory V OŠConnor, Brian Leavy, and Murat Yilmaz. 2015. Exploring the relationship between software process adaptive capability and organisational performance. *IEEE Transactions on Software Engineering* 41, 12 (2015), 1169–1183.

[14] Simone Colombo, Salman Nazir, Davide Manca, et al. 2013. Virtual reality as effective tool for training and decision-making: preliminary results of experiments performed with a plant simulator. In *European HSE Conference and Exhibition*. Society of Petroleum Engineers.

[15] António Pedro Costa, Luis Paulo Reis, and Maria João Loureiro. 2014. Hybrid user centered development methodology: An application to educational software development. In *International Conference on Web-Based Learning*. Springer, 243–253.

[16] Lars Donath, Roland Rössler, and Oliver Faude. 2016. Effects of virtual reality training (exergaming) compared to alternative exercise training and passive control on standing balance and functional mobility in healthy community-dwelling seniors: a meta-analytical review. *Sports medicine* 46, 9 (2016), 1293–1309.

[17] Yihai Fang and Jochen Teizer. 2014. A Multi-user Virtual 3D Training Environment to Advance Collaboration Among Crane Operator and Ground Personnel in Blind Lifts. In *Computing in Civil and Building Engineering (2014)*. 2071–2078.

[18] Ulas Gulec, Murat Yilmaz, and Veysi Isler. 2017. A Literature Survey: Is it Necessary to Develop a New Software Development Methodology for Virtual Reality Projects? *Journal of Universal Computer Science* 23, 8 (2017), 725–754.

[19] Thomas Hainey, Thomas M Connolly, Mark Stansfield, and Elizabeth A Boyle. 2011. Evaluation of a game to teach requirements collection and analysis in software engineering at tertiary education level. *Computers & Education* 56, 1 (2011), 21–35.

[20] Ilona Heldal. 2007. Supporting participation in planning new roads by using virtual reality systems. *Virtual Reality* 11, 2-3 (2007), 145–159.

[21] Watts S Humphrey. 1989. The software engineering process: definition and scope. *ACM SIGSOFT Software Engineering Notes* 14, 4 (1989), 82–83.

[22] Michelle R Kandalaft, Nyaz Didehbani, Daniel C Krawczyk, Tandra T Allen, and Sandra B Chapman. 2013. Virtual reality social cognition training for young adults with high-functioning autism. *Journal of autism and developmental disorders* 43, 1 (2013), 34–44.

[23] Johanna Kollmann, Helen Sharp, and Ann Blandford. 2009. The importance of identity and vision to user experience designers on agile projects. In *Agile Conference, 2009. AGILE'09*. IEEE, 11–18.

[24] Gonzalo Lorenzo, Asunción Lledó, Jorge Pomares, and Rosabel Roig. 2016. Design and application of an immersive virtual reality system to enhance emotional skills for children with autism spectrum disorders. *Computers & Education* 98 (2016), 192–205.

[25] Begoña Losada, Maite Urretavizcaya, and Isabel Fernández-Castro. 2013. A guide to agile development of interactive software with a ŞUser ObjectivesŤ-driven methodology. *Science of Computer Programming* 78, 11 (2013), 2268–2281.

[26] Begoña Losada, Maite Urretavizcaya, Juan-Miguel López-Gil, and Isabel Fernández-Castro. 2012. Combining InterMod agile methodology with usability engineering in a mobile application development. In *Proceedings of the 13th International Conference on Interacción Persona-Ordenador*. ACM, 39.

[27] Mark Lycett, Robert D Macredie, Chaitali Patel, and Ray J Paul. 2003. Migrating agile methods to standardized development practice. *Computer* 36, 6 (2003), 79–85.

[28] Steve McConnell. 2001. Who needs software engineering? *IEEE Software* 18, 1 (2001), 5–8.

[29] Zahira Merchant, Ernest T Goetz, Lauren Cifuentes, Wendy Keeney-Kennicutt, and Trina J Davis. 2014. Effectiveness of virtual reality-based instruction on students' learning outcomes in K-12 and higher education: A meta-analysis. *Computers & Education* 70 (2014), 29–40.

[30] David Parsons and Rosemary Stockdale. 2010. Cloud as context: Virtual world learning with open wonderland. In *Proceedings of the 9th World Conference on Mobile and Contextual Learning, Malta*. 123–130.

[31] Joseph Psotka. 1995. Immersive training systems: Virtual reality and education and training. *Instructional science* 23, 5 (1995), 405–431.

[32] M Rizwan Jameel Qureshi and SA Hussain. 2008. An adaptive software development process model. *Advances in Engineering Software* 39, 8 (2008), 654–658.

[33] Guillermo Rodriguez, Álvaro Soria, and Marcelo Campo. 2015. Virtual Scrum: A teaching aid to introduce undergraduate software engineering students to scrum. *Computer Applications in Engineering Education* 23, 1 (2015), 147–156.

[34] Eran Rubin and Hillel Rubin. 2011. Supporting agile software development through active documentation. *Requirements Engineering* 16, 2 (2011), 117–132.

[35] Adrian Rusu, Robert Russell, Edward Burns, and Andrew Fabian. 2011. Employing software maintenance techniques via a tower-defense serious computer game. *Edutainment Technologies. Educational Games and Virtual Reality/Augmented Reality Applications* (2011), 176–184.

[36] Alcínia Z Sampaio and Octávio P Martins. 2014. The application of virtual reality technology in the construction of bridge: The cantilever and incremental launching methods. *Automation in construction* 37 (2014), 58–67.

[37] Neal E Seymour, Anthony G Gallagher, Sanziana A Roman, Michael K OŠbrien, Vipin K Bansal, Dana K Andersen, and Richard M Satava. 2002. Virtual reality training improves operating room performance: results of a randomized, double-blinded study. *Annals of surgery* 236, 4 (2002), 458.

[38] Osama Sohaib and Khalid Khan. 2010. Integrating usability engineering and agile software development: A literature review. In *Computer design and applications (ICCDA), 2010 international conference on*, Vol. 2. IEEE, V2–32.

[39] Frank Tsui, Orlando Karam, and Barbara Bernal. 2016. *Essentials of software engineering*. Jones & Bartlett Learning.

[40] Lex Van Velsen, Jobke Wentzel, and Julia EWC Van Gemert-Pijnen. 2013. Designing eHealth that matters via a multidisciplinary requirements development approach. *JMIR research protocols* 2, 1 (2013).

[41] En Ye, Chang Liu, and Jennifer A Polack-Wahl. 2007. Enhancing software engineering education using teaching aids in 3-D online virtual worlds. In *Frontiers in education conference-global engineering: knowledge without borders, opportunities without passports, 2007. FIE'07. 37th annual*. IEEE, T1E–8.

[42] Eugenia Yiannakopoulou, Nikolaos Nikiteas, Despina Perrea, and Christos Tsigris. 2015. Virtual reality simulators and training in laparoscopic surgery. *International Journal of Surgery* 13 (2015), 60–64.