Usability of the EMERGO player environment for scenario-based serious games

Usability of the EMERGO player environment for scenario-based serious games

Aad Slootmaker
Open University of the Netherlands, Heerlen, The Netherlands
aad.slootmaker@ou.nl

Rob Nadolski
Open University of the Netherlands, Heerlen, The Netherlands

Hub Kurvers
Open University of the Netherlands, Heerlen, The Netherlands

Hans Hummel
Open University of the Netherlands, Heerlen, The Netherlands

Rob Koper
Open University of the Netherlands, Heerlen, The Netherlands

Abstract

The EMERGO online platform enables the development and delivery of scenario-based serious games that foster students in acquiring professional competence. One of the main goals of the platform was to provide a user-friendly player environment that enables students to learn complex cognitive skills in authentic professional settings. We present the results of a combined quantitative and qualitative study of the usability of the platform´s player environment. We used questionnaires, made notes and group discussions with 167 students who played two games about ICT management. The analysis shows that the usability of the player environment is adequate, which is lower than found in previous more superficial studies of the environment. Its operability is mostly valued, but its understandability and user interface aesthetics are somewhat problematic, probably because students got inadequate instruction beforehand and expected a more realistic video game interface.

Keywords

Functionality; operability; performance efficiency; player environment; reliability; scenario-based; serious games; software quality; System Usability Scale; understandability; usability, user error protection, user interface aesthetics.

Introduction

Serious games (SGs) are powerful vehicles to provide learning in a more attractive, intense and challenging way and are still of growing importance in education, e.g., to learn complex cognitive skills in authentic professional settings. These skills involve mental processes that occur in the mind while using, transforming or supplementing available knowledge, which involves higher-order activities like problem solving, reasoning, thinking, assessing and concluding.

The uptake of SGs is still hampered, because their development requires high technical demands, high costs, and high time investment, which also involves testing all possible student paths. In addition, the field lacks a good architecture for SG development (Nadolski, Hummel, Slootmaker, & Van der Vegt, 2012), standards for SG design (Klemke et al., 2015), including usability design, and standardized ways of evaluation, including usability evaluation, also during development. Furthermore measuring usability itself is complex, as Lewis (2014) emphasizes: "The measurement of usability is complex because usability is not a specific property of a person or thing. You cannot measure usability with a simple 'usability' thermometer. Rather, it is an emergent property dependent on interactions among users, products, tasks, and environments.".

To partly overcome above problems related to SG development our institution developed the EMERGO online platform for development and delivery of SGs (Nadolski et al., 2008). It should reduce high technical demands by developing generic adaptable components and reduce high costs and time investment by reusing these components in other SGs, also for different content domains. One of the main goals was to provide an intuitive and immersive player environment that enables students to perform authentic tasks. This environment currently offers twenty nine components that support different (didactical) functions that may be present in SGs. In addition, the platform offers environments to author games, to monitor students, and to manage users and game runs.

The research goal of this study is to evaluate the usability of the EMERGO player environment in detail. We already evaluated the platform's authoring environment in detail which showed its understandability and learnability to be problematic, operability to be somewhat problematic, and functionality and reliability to be valued (Slootmaker, Hummel, & Koper, 2017). However, the player environment, although being essential for the platform, has not been evaluated in detail yet. Although superficial evaluations with less available components (Nadolski et al., 2008; Slootmaker, Kurvers, Hummel, & Koper, 2014) show that students are satisfied till very satisfied about using the environment, we question if users are still satisfied, are satisfied about newly developed components and if and why components differ in usability. We present the results of a combined quantitative and qualitative evaluation of the usability of the player environment.

We first give background information on usability in the 'Background' section. In the 'EMERGO' section we present EMERGO, its player environment and available components. In the 'Method' section we explain the method that is followed in order to arrive at our results and findings in the 'Results and findings' section. Finally, in the 'Conclusion and discussion' section, we present the main conclusions to be drawn from this study.

Background

Although usability is a very important quality factor of a software system, no single definition of usability exists which takes into account all of its possible aspects (Dubey & Rana, 2010). Nielsen (1993), for instance, defined usability by its quality of five components: learnability (for novice

users), efficiency (amount of time to accomplish task), memorability (for frequent users), errors (number, severity, recoverability), and satisfaction (pleasantness). ISO/IEC (2011), on the other hand, defined usability as the degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use. More recent concepts are user-centered design, which implies involving end-users in each stage of the design process, and user experience, which involves the effects of usability factors, usefulness factors (how useful is a tool for a task), and emotional impact (broader than Nielsen's satisfaction), and strongly depends on the context of a usage by a particular user (Hartson & Pyla, 2012).

Apart from a lack of a single usability definition, there is also no single method to measure it. The most widely used standardized usability questionnaires for assessment of usability are the QUIS, the SUMI, the PSSUQ and the SUS (System Usability Scale; Brooke, 1996, 2013; Sauro, 2011), of which the SUS, with a global reliability of 0.92, is probably most widely used (Lewis, 2014). The SUS consists of 10 items with alternating positive and negative tone, produces one usability score, allows for interpretation of its data in a normative way, and seems to tolerate minor changes to its wording, e.g., 'game' instead of the original 'system' (Lewis, 2014). A disadvantage of above questionnaires is that they produce a general score (SUS) or scores on general usability aspects (QUIS, SUMI and PSSUQ), which makes them less appropriate to identify more specific interface related usability issues.

Usability is a decisive success factor for video games, which is illustrated by the large volume of studies on usability, and playability, of games. The concept of playability is broader than usability and is defined as "the degree to which a game is fun to play and is usable, with an emphasis on the interaction style and plot-quality of the game; the quality of game play" (Usability-First 2017). Playability may be affected by the quality of the storyline, the degree of responsiveness, intensity of interaction, pace, control, intricacy, customizability, realism, and social and team support, and the quality of graphics and sound. A good playability will improve the player experience by providing an immersive and challenging environment where a player will experience enjoyment and tension, but also disappointment and frustration (Mekler et al., 2014). Federoff (2002) compiled a list of game usability heuristics that can be used for video game creation and evaluation, and classifies them into three areas: game interface, game mechanics (fostering game rules and interactivity), and game play (problems and challenges a player must face). The game interface and game mechanics areas cover usability aspects like, e.g., interface consistency, while the game play issues cover more typical playability aspects like, e.g., a variable difficulty level.

For SGs the same usability and playability aspects play a role as for video games, but because their main goal is learning, which should be in good balance with fun (Franzwa et al., 2014), aspects that support learning obviously need more attention. According to Ibrahim et al. (2014) SGs aim to motivate learners, to give them appropriate feedback, to improve their skills at the right level, and to improve collaboration within groups. The author compiled a list of playability guidelines to evaluate and enhance the playability of SGs. The guidelines, which mostly fall into Federoff's game play area, are classified into twelve categories that cover all SG aspects: game goals (playful and educational), balanceability (between learning and fun), game challenge (e.g., not too difficult or easy), feedback (e.g., to understand why one has failed), interactivity (e.g., clear and simple instructions and rules), adaptation (e.g., to the individual pace of the player), game control (e.g., a player should be in control), ethics, realism (e.g., a realistic scenario), game reward, structuring (e.g., offer defined tasks and subtasks), and player knowledge (activate and use players' prior knowledge and skills). According to Hamari et al. (2016), both engagement and challenge in game-based learning have a positive effect on learning. Challenge is a strong predictor of learning

outcomes, but should be adapted to the learner's growing skills in order to support continued learning.

A player environment for SGs (or video games) is only partly responsible for the usability and playability of played games. Game usability indeed depends on the usability of the environment's different components. However, game playability for a large part depends on game design and content, e.g., the quality of the storyline, feedback or graphics and sound, which cannot be influenced by a player environment. Of course the environment should support playability aspects like responsiveness and intensity of interaction, and should enable to play games that conform to playability guidelines.

As playability for a player environment for SGs mostly depends on game play, we focus on its usability. We use the definition by ISO/IEC 25010:2011 (ISO/IEC, 2011), where usability is one out of eight software quality characteristics. The seven other characteristics are functionality, reliability, performance efficiency, compatibility, security, maintainability and portability. Usability is further subdivided into six aspects: understandability, learnability, operability, user error protection, user interface aesthetics and accessibility. Note that for better readability we replace ISO/IEC characteristic functional suitability by functionality and appropriateness recognizability by understandability.

Not many authors evaluated the usability of player environments for SGs. Gaeta et al. (2014) evaluated the usability of a developed Storytelling Complex Learning Object (SCLO) on the IWT (Intelligent Web Teacher) e-learning platform and found the mean SUS score to be 64.13 which corresponds to a user-friendliness between "ok" and "good" (Bangor, Kortum, & Miller 2008).

EMERGO

We developed the *EMERGO method* and *online platform* (Nadolski et al., 2008) to simplify and better support the development and delivery of scenario-based SGs. In this kind of games learners are confronted with realistic ill-defined problems, often allowing multiple solutions and requiring application of necessary methodologies or tools and collaboration with fellow learners (Westera, Nadolski, Hummel, & Wopereis, 2008). The scenario describes the problem space and how it should adapt to the student's actions. The platform currently offers twenty nine generic components that support different (didactical) functions that may be present in scenario-based SGs and offers environments to play games, to author games (Slootmaker, Hummel, & Koper, 2017), to monitor students, and to manage users and game runs. EMERGO has been used to develop twenty six games for all kind of disciplines and supports acquiring four out of five kinds of learned capabilities as defined by Gagné (1985): intellectual skills, cognitive strategies, verbal information, and attitudes. Motor skills are not (yet) supported. The online platform is Open Source and is available on SourceForge (EMERGO, 2017).

EMERGO games are developed by a multidisciplinary team that consists of content matter experts, educational technologists, interaction designers and ICT developers, which may be temporarily reinforced with other specialists, if needed (e.g., for video production). After agreeing on a global description of the game, the team writes the game scenario in three steps, where every step adds more detail. In the end, the scenario describes all tasks to be done and why, when and where, and in what order. It also describes which PC's (Playing Characters) and NPC's (Non-Playing Characters) are involved in a task, which materials and tooling are needed, when the task is completed and how this is assessed, and which feedback is given and when, in what form and by

whom. The authoring environment is used to convert the scenario and materials into game content that can be previewed and tested in the player environment.

In the following sections, we will describe the playing of EMERGO games, the EMERGO player environment and its generic components.

Playing EMERGO games

In a typical EMERGO game a student enters an authentic environment where he works as a trainee. He can navigate to different locations where he finds NPC's like his supervisor, colleagues, experts or specialists, or can attend interviews or meetings (see Figure 1). In the environment he has a tablet with apps, e.g., a task overview (see Figure 2), a resources app, an (in-game) email app or an app to conduct tests. He also has a memo recorder to record interesting parts of interviews and meetings, and a notepad to make contextualized notes.

The student gets tasks from his supervisor or other NPC's, either in person or by email, and can send in his outcomes by email, either to NPCs or to PCs (fellow students or educators). He can be assessed on every action he performs, e.g., which interviews he attends, which questions he asks, which resources he consults or which mails he reads or sends. In addition, he can be assessed using tests that enable, e.g., measuring foreknowledge or performance. Depending on his actions or progress game script may adapt the environment on micro level by, e.g., (un)locking locations, sending a mail, showing an alert, changing a NPC reaction, releasing new resources or new interview questions, or on macro level by, e.g., providing new or alternative tasks. The student gets feedback on his performance by NPC's, in person, by mail or as screen text, or in tests. This feedback can incorporate mail attachments or release of resources such as worked out examples or expert reports. If an educator has a PC role, he can give students feedback within the game, otherwise he can give feedback by impersonating an NPC. The student gets navigation support through alerts, e.g., reminders for meetings or instructions where to go next.

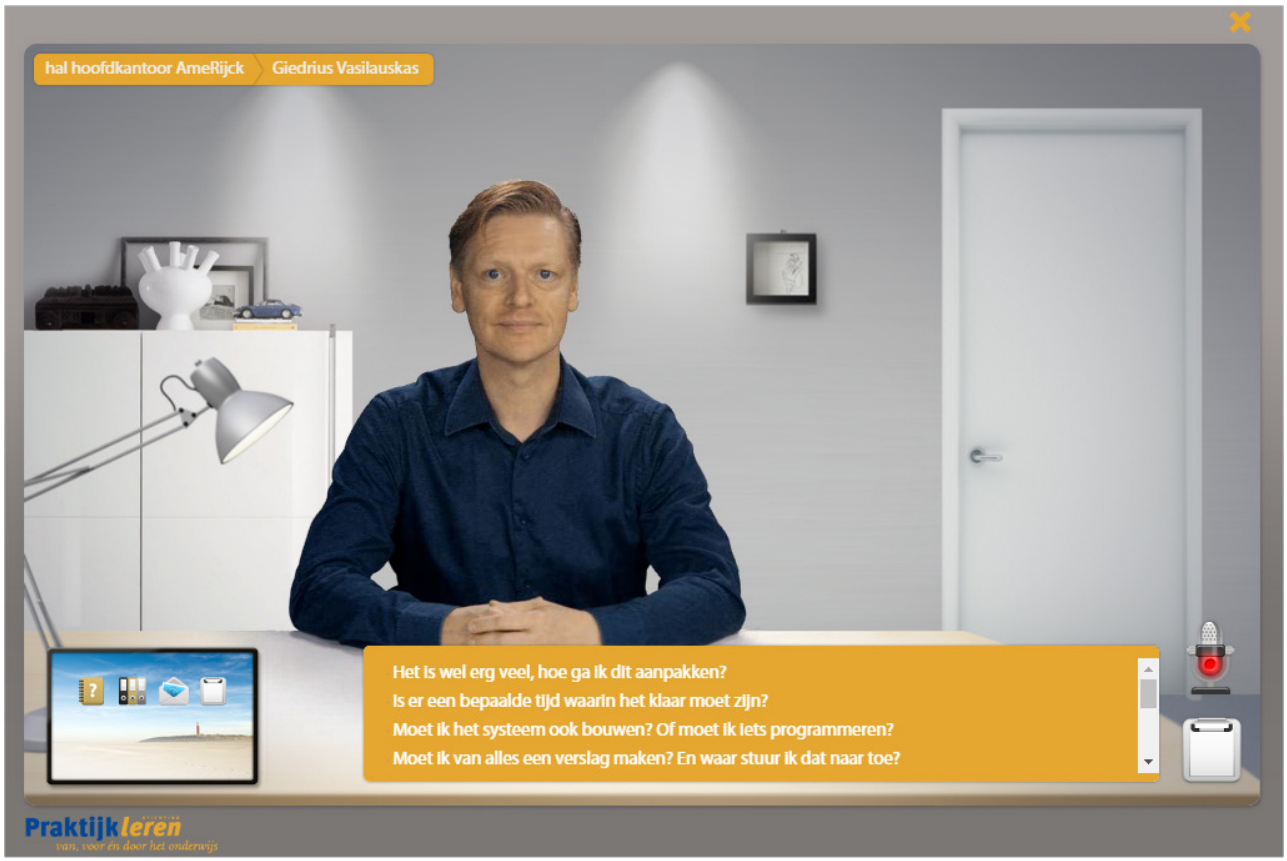

Figure 1. An interview with a supervisor in a game skin developed for Stichting Praktijkleren. Bottom left we see the *tablet* icon and bottom right we see the *memo recorder* and *notepad* icons

The EMERGO player environment

Our main initial goal was to develop an intuitive, immersive, reliable and stable online player environment that might be adapted according to the actions and progress of a student. The environment should support multiple game roles, should offer a set of generic, reusable, adaptable, and loosely coupled components that support different (didactical) functions, and should save all student actions, for game script to operate on, and for evaluation and research purposes. The environment should enable ICT developers to rather easily add new generic components, by applying a generic component template. The current environment also supports the use of skins and plugins. Skins (currently four) make it possible to offer (external) parties their own look and feel, and plugins enable ICT developers to build or add specific components that are only expected to be needed once, e.g., for experiments, or for integrating, e.g., Google forms or the Unity web player.

The player environment and its components support most heuristics and guidelines compiled by Federoff (2002) and Ibrahim et al. (2014). However, adding user-generated content is only allowed in a few components, reversal of actions is not supported, and games are not re-playable without help of an admin. The environment offers a powerful option for previewing game and game components which allows for immediately inspecting if content is entered correctly and for testing the game scenario or playability. It is possible to make recordings till a certain point in time and to use these recordings as starting points to test specific parts of the scenario. These recordings can also simply be shared with others, e.g., for demo's. In addition, multiple instances of the environment can be previewed simultaneously which allows for testing collaboration between

students. Another powerful option is that students' progress can be adjusted by either persons or web services. In case a student is stuck, e.g., by a door that stays locked, an educator or admin can open the environment using a student's progress, diagnose the problem and subsequently fix it by live adjusting student's progress. It is also possible to adjust the progress for all students in a game run simultaneously, which allows for, e.g., live making available certain tasks for all students simultaneously. Student's progress may also be read or adjusted using web services, which allows for exchanging progress data with other applications that may adapt the player environment for a certain student this way. For instance, a student's mood determined by analysis of webcam data may result in another type of support in the game.

The player environment's generic components

Table 1 shows all currently available player environment components, their description and which different (didactical) functions they support. The eight functions that may be present in scenario-based SGs are: present and adapt the environment (E); assign tasks and provide task overview (T); present knowledge (K); assess learner (A); provide feedback (F); support processing of information (P); support collaboration (C); and support navigation (N). Note that one component may serve several functions and that one function may involve several components. For instance, the *conversations* component can be used to assign a task, to present knowledge or to provide feedback. And the *script* component has to assess the learner to trigger the *conversations* component to give the right feedback.

Table 1. Player environment components and their possible functions

Component	**Description**	**Functions**
Navigation	Enable spatial navigation through the game	E
Conversations	Enable communication with NPC's using video or text	ETKF
Notepad	Enable making contextualized notes	EP
Memo recorder	Enable recording of conversations	EP
Alerts	Provide popup texts	EFN
Notifications	Provide (accumulated) embedded texts	EFN
Scores	Provide score overview	EF
Profile	Enable sharing profile with PC's	EC
Chat	Enable communication with PC's	EC
iSpot	Enable webcam recordings as reaction on (patient) videos. Recordings may be assessed by educators or fellow students	EAFC
Tablet	Enable choosing apps	E
Tasks	Provide task (completion) overview. App	ET
Resources	Enable consulting resources. App	EKF
Email	Enable communication with NPC's and between PC's. App	ETKFC
Assessments	Enable conducting tests. App	EAF
Logbook	Provide overview of notes. App	EP
Memo player	Enable playing back of recordings. App	EP
Google maps	Enable inspecting maps with markers. App	EK
Directing	Enable analyzing communication between NPC's. App	EP
Drag drop forms	Enable categorizing of content elements using drag and drop. App	EKF
Input forms	Enable filling in forms with different input controls. App	EKF

Graphical forms	Enable presenting and positioning of graphical elements. App	EKF
Text fragment selector	Enable selecting of relevant text fragments in a text. App	EKF
Video scene selector	Enable categorizing of scenes in a video. App	EKF
Tests	Enable doing psychological testing. App	EK
Game manual	Provide help on game interface. App	EN
Items	Contains questions to be used in the Assessments component	EAF
States	Contains states to be used by the Script or Scores component	A
Script	Contains rules to assess the learner and adapt the game on micro and macro level	ETKAFP CN

Within the player environment the *navigation* component renders the different locations and background objects, and enables navigation by clickable interface elements, e.g., doors. The *conversations* till *tablet* components are presented on top of a location and may be present on all locations or specific ones. The *tablet* is opened on top of *conversations* and is used to present the *tasks* till *tests* tablet apps. The *items* till *script* components are no recognizable entities in the player environment and are either used by other components or to adapt the environment. Most components allow for having multiple instantiations in the player environment, which enables thematically arranging game content, e.g., one *conversations* component per interviewee or one *resources* app per type of resource. In case of multiple game roles, components may be allocated to specific game roles, which allows for a different environment per game role.

The operation of all components is expected to be rather easy. The component interfaces do not present complex concepts, structures or dynamics (Murray, 2004), so we do not expect to find related problems. However, how the (didactical) functions are translated into usable interfaces may leave room for improvement, which is the motive for our usability evaluation.

Figure 2. The *tasks* app on the *tablet*

Method

To evaluate the EMERGO player environment we choose two games on IT administration (in Dutch), that have been developed, in 2014, by the Dutch Foundation for Practice-based Learning (Stichting Praktijkleren), in the context of the SLEM project. In both games students have to develop an information system, going through five generally accepted phases for solving IT-problems, which include writing a functional design, a technical design and a test plan. However, in the second game the system to be developed is more complex and given support is less substantive than in the first game.

Both games are developed using the EMERGO method and platform, are typical examples of EMERGO games, and use the same player environment components, sixteen out of twenty two available at the time: the *navigation*, *conversations*, *notepad*, *memo recorder*, *alerts*, *tablet*, *tasks*, *resources*, *email*, *assessments*, *logbook*, *memo player*, *game manual*, *items*, *states* and *script* component.

Participants

167 secondary vocational education students in IT administration from four Dutch Regional Centers for Vocational Education (ROCs) participated in this research. These students (two female and 165 male; mean age 19.3 years) were all in their second year of study. Students played either game 1 or game 2. Game 1 was not yet embedded in the curriculum and was tested by 86 students, while game 2 was used in regular education by 81 students. The survey was conducted in the fall of

2014 and the first half of 2015. Apart from usability, game content quality and learning effects (Nadolski, & Hummel, 2016) were evaluated.

Data collection method

We opted for a combination of quantitative and qualitative research to both get hard as well as more subjective usability data for the two games, and thus for the player environment and its components.

At the start of a game students filled in a pre-questionnaire (in Dutch) that included three MC questions (interval, 10-point scale) to determine their prior ICT skills, entertainment game skills and SGs skills, in order to find out whether prior skills would affect the experienced usability.

During a game, students were asked to write down points of improvement (tips) and points of satisfaction (tops), regarding both operation and content of the game. Present educators also kept a record of oral comments made by students and of problems they encountered, which were divided into tips and tops.

At the end of a game students filled in a post-questionnaire (in Dutch) that included ten MC questions to determine the SUS mean score (ordinal, 5-point Liker scale ranging from *strongly disagree* to *strongly agree*), five general usability related MC questions, and eleven player environment component specific MC questions (ordinal, the same 5-point scale as for the SUS questions). In addition, students were asked to give a final grade (interval, 10-point scale) for the operation of the game and to again fill in tips and tops.

Both game sessions were followed by a group discussion (in Dutch) with involved students and educators, again about tips and tops.

Data analysis

Our quantitative analysis is based on the data of 120 students out of the original 167. 47 students fell of, because they did not fill in the post-questionnaire at all or not seriously. The latter could be identified because students choose the same value for all SUS questions while values are expected to fluctuate because of the questions' alternating positive or negative tone. 56 students completed game 1 (non-response 35%) and 64 students completed game 2 (non-response 21%). Next we calculated the mean SUS scores (Sauro, 2011), and means and standard deviations for the MC questions, and correlations between the different kind of data, separately for game 1 and 2, and in total. We used either the Pearson product momentum correlation or Spearman's rank correlation, depending on the data types involved, being interval or ordinal.

For the qualitative data analysis we used data of all 167 students, whether they completed the game or not. We assembled all tips and tops, 637 in total. We left out 280 tips and tops, because they were unclear or related to the game content, which is beyond the scope of this publication. 357 tips and tops were related to game operation, 195 tips and 162 tops. Next we identified 58 unique tips and 24 unique tops and their frequency, and related them to either the player environment in general or to specific player environment components, and to usability aspects or other ISO/IEC software quality characteristics.

Results and findings

We present the quantitative results and qualitative findings related to our original evaluation goal which was to evaluate the usability of the EMERGO player environment in detail. Results and

findings are based on two games played within the environment. Results are presented for both games separately and in total, and findings are presented in total.

Quantitative results

Table 2 shows the SUS mean score for the games to be 58.10, which is below the mean score of 68.05 for web pages and applications as determined by Bangor, Kortum, and Miller (2008). A SUS score of 58.10 corresponds to a user-friendliness between "ok" (52.01) and "good" (72.75). The mean SUS score for game 2 is lower than for game 1 but not significantly.

Table 2: Mean SUS scores

Game 1 (n = 56)		**Game 2 (n = 64)**		**Total (n = 120)**	
M	SD	M	SD	M	SD
62.10	12.80	54.60	14.90	58.10	14.40

Table 3 shows the results for the five general usability related questions that give an indication of the general operability and understandability of the games. All total mean values are within 3 (*neutral*) and 4 (*agree*) and do not differ significantly. The mean value for all five questions is 3.65, so somewhat closer to *agree* than to *neutral*, and is equal for game 1 and 2.

Table 3: General usability

Question	**Usability aspect**	**Game 1 (n = 56)**		**Game 2 (n = 64)**		**Total (n = 120)**	
I could start and shut down the game without any problems	Operability	M	SD	M	SD	M	SD
		3.79	1.33	3.75	1.14	3.77	1.23
I think I have sufficient control within the game	Operability	M	SD	M	SD	M	SD
		3.21	1.12	3.45	1.01	3.34	1.07
I always know where I'm in the game	Operability	M	SD	M	SD	M	SD
		3.66	1.01	3.69	0.94	3.68	0.97
The space metaphor as a basis for the design of the game is clear	Understandability	M	SD	M	SD	M	SD
		4.04	1.04	3.95	0.88	3.99	0.96
I find the operating instructions in the game clear	Understandability	M	SD	M	SD	M	SD
		3.54	0.99	3.41	0.94	3.47	0.96
Mean		**M**	**SD**	**M**	**SD**	**M**	**SD**
		3.65	0.31	3.65	0.22	3.65	0.25

Table 4 shows the results for the eleven component specific questions that give an indication of the component specific operability and understandability. All total mean values are within 3 (*neutral*) and 4 (*agree*) and do not differ significantly. The mean value for all eleven questions is 3.65, so somewhat closer to *agree* than to *neutral*. The mean value for game 2 is somewhat lower than for game 1 but not significantly. Note that the question "How I had to use the assessments was clear" was only asked for game 1 and that no questions were asked about the *notepad* and *logbook* component.

Table 4: Component specific usability, MV = Missing Value, NA = Not Applicable

Question	Component	Game 1 (n = 56)		Game 2 (n = 64)		Total (n = 120)	
How I had to navigate within the game was clear	Navigation	M 3.86	SD 1.03	M 3.53	SD 0.93	M 3.68	SD 1.11
How I had to conduct conversations and interviews within the game was clear	Conversations	M 4.09	SD 0.82	M 3.30	SD 1.20	M 3.67	SD 1.23
How to record conversations and interviews was clear	Memo recorder	M 3.20	SD 1.30	M 3.34	SD 1.07	M 3.28	SD 1.18
The operation of popup notifications was clear	Alerts	M 3.75	SD 1.07	M 3.41	SD 0.92	M 3.57	SD 1.00
How to use the tablet was clear	Tablet	M 3.98	SD 1.02	M 3.77	SD 0.89	M 3.87	SD 0.95
How to use the task overview was clear	Tasks	M 3.70	SD 1.20	M 3.58	SD 0.99	M 3.63	SD 1.09
How to use the resources was clear	Resources	M 3.79	SD 1.12	M 3.61	SD 0.92	M 3.69	SD 1.02
How to use the email was clear	Email	M 4.05	SD 0.90	M 3.61	SD 1.05	M 3.82	SD 1.00
How I had to use the assessments was clear	Assessments	M 3.57	SD 1.16	M MV	SD MV	M NA	SD NA
How I could play back conversations and interviews was clear	Memo player	M 3.89	SD 1.11	M 3.56	SD 1.05	M 3.72	SD 1.09
How to use the game manual was clear	Game manual	M 3.64	SD 1.09	M 3.58	SD 0.96	M 3.61	SD 1.01
Mean		**M** 3.77	**SD** 0.25	**M** 3.53	**SD** 0.14	**M** 3.65	**SD** 0.15

The final player environment/game operation grade is 6.23, see table 5. In the Dutch language area a grade of 6 corresponds to adequate. The final game operation grade for game 2 is somewhat lower than for game 1 but not significantly.

Table 5: Final game operation grade

Question	Game 1 (n = 56)		Game 2 (n = 64)		Total (n = 120)	
Enter a final grade for the game operation	M 6.32	SD 2.05	M 6.14	SD 2.02	M 6.23	SD 2.03

We found no correlation ($\rho <= 0.1$) between the prior skills (ICT skills, entertainment game skills and SGs skills) and the mean SUS scores, the general usability data, the component specific usability data and the final game operation grades. The correlations between these latter four types of data all are positive, significant ($p < 0.01$) and weak ($0.1 < \rho <= 0.3$), moderate ($0.3 < \rho <= 0.5$) or strong ($\rho > 0.5$), so the data seems to be internally consistent.

Qualitative findings

In general, students have strong opinions about the games and can be either very positive or very negative.

Table 6 shows tips and tops, and their numbers, combined for general usability aspects and software qualities. *Operability* is mostly valued, but a number of students miss feedback or support. *Understandability* is somewhat problematic because of missing instruction and tips, although some students are very positive. *User interface aesthetics* is somewhat problematic because students expect a more realistic 3D environment, although others are positive about the interface. *Performance efficiency* is problematic, which was mainly caused by slow wireless internet connections in some ROCs. *Functionality* is little problematic because of disturbing alerts and a missing timeline. *Reliability* is mostly valued, but some students miss better browser support. We could not relate tips or tops to other usability aspects, namely learnability, user error protection or accessibility, or other software quality characteristics, namely compatibility, security, maintainability, and portability.

Table 6: Tips and tops for general usability aspects and software qualities

Aspect	**Tips**	**N**	**Tops**	**N**
Operability	Miss feedback on download status and errors, sounds when clicking or better support using arrow keys	11	Game operation is good and easy	80
Understandability	Miss instruction and tips	11	Game is very clear and realistic	4
User interface aesthetics	Would like to have a more realistic and dynamic, and less structured and artificial environment where you can walk around yourself in 3D instead of clicking on doors, and communicate through talking instead of choosing predefined questions	16	Game is beautifully made, layout is good and it has a nice interface	9
Performance efficiency	Slow page loading and game operation, rendering problems, problems while playing video files and uploading files, no response on user actions, and progress being not or incorrectly saved	72	-	-
Functionality	Find the alerts during interviews to be disturbing and miss a timeline to indicate your progress	4	-	-
Reliability	Miss better browser support	3	It works fine and as expected	8

Table 7 shows tips and tops, and their numbers, combined for component specific usability aspects and software qualities. The *navigation*'s *operability* is mostly valued, although some students found the parallax effect (horizontally scrolling in a hallway with doors) to be unhandy. The *conversations*' *operability* is mostly valued, although a number of students miss video controls, which, however, are left out deliberately for more realism. Its *user error protection* is somewhat problematic, because leaving a location will end a conversation without any warning. Its *functionality* is little problematic, because predefined questions may not cover all questions one would like to ask. The *notepad*'s *operability* and *functionality* are somewhat problematic, because

it cannot be dragged and is not always available, and because some students do not see any advantage of making context specific notes, although others value it. The *tablet*'s *operability* is valued. The *memo recorder*'s, *tablet*'s and *tasks*' *functionality* is also valued. The *resources*' *functionality* is little problematic, because clicking a link resulted in two file downloads. The *email*'s *reliability* is somewhat problematic, because the input control for mail text was not rendered in all browsers, although some students value its *operability*. The *assessments*' *operability* is somewhat problematic, because some students find finishing of an assessment to be cumbersome. The *logbook*'s *functionality* is little problematic, because some students find the presentation of notes to be unclear. The *memo player*'s *functionality* is little problematic, because some students miss the original background. We could not relate tips or tops to the *alerts* and *game manual*, or to other usability aspects, namely understandability, learnability, user interface aesthetics and accessibility, or other software quality characteristics, namely performance efficiency, compatibility, security, maintainability and portability.

Table 7: Tips and tops for component specific usability aspects and software qualities

Component	**Aspect**	**Tips**	**N**	**Tops**	**N**
Navigation	Operability	Find the parallax effect to be unhandy	3	Navigation is good, clear and easy	15
Conversations	Operability	Miss video controls for play back and forward, and replay	15	The intuitive way you ask questions is good	18
	User error protection	A click on a door ends a conversation	9	-	-
	Functionality	Disadvantage of predefined questions is that questions may be missing	1	The possibility to ask interview questions in your own order	1
Notepad	Operability	Cannot drag it	9	-	-
	Functionality	It is not always available and don't see the advantage of making context specific notes instead of general notes	13	The possibility to make notes is good and very handy	6
Memo recorder	Functionality	-	-	The recording function is very handy	3
Alerts	-	-	-	-	-
Tablet	Operability	-	-	Easy and clear	3
	Functionality	-	-	The tablet works fine	4
Tasks	Functionality	-	-	Task overview is important and necessary, and handy to see were you are and what you have done	8
Resources	Functionality	Clicking a link results in two file downloads	9	-	-
Email	Operability	-	-	Mail works fine	4

	Reliability	The input control for mail text is not rendered in some browsers	7	-	-
Assessments	Operability	Find finishing of an assessment to be cumbersome	6	-	-
Logbook	Functionality	Present notes more clearly	4	-	-
Memo player	Functionality	The original background is missing	2	-	-
Game manual	-	-	-	-	-

Conclusions and discussion

Our research goal was to evaluate the usability of the EMERGO player environment for scenario-based serious games in detail. To accomplish this goal, we evaluated the usability of two games about ICT management. Although both games used only sixteen player environment components out of twenty two available at the time, we have strong indications that our results and findings are representative for the usability of the player environment as a whole. First, both games have used most components (73%) and are typical examples of EMERGO games, meaning that the components have been used in multiple games before. Second, the components that were not used are less used in other games before and have comparable interfaces, input controls, and complexities as the used ones.

In general we find the usability of the player environment to be adequate. Quantitative analysis shows that the mean SUS score is 58.10, which corresponds to a user-friendliness between “ok” and “good” (Bangor, Kortum, & Miller, 2008), the mean given grade for operation is 6.23, which corresponds to adequate, and most students agree that general operability and understandability are ok (mean 3.65, in between neutral and agree. The found mean SUS score is lower than the mean overall score of 68.05 for web pages and applications (Bangor, Kortum, & Miller, 2008) and lower than the score of 64.13 as found by Gaeta et al. (2014), who evaluated the usability of a comparable e-learning environment. Qualitative analysis indicates that usability aspect operability is mostly valued, but that understandability and user interface aesthetics are somewhat problematic (e.g., missing instruction and more realistic environment). Software quality characteristic performance efficiency was problematic at some institutions, functionality was little problematic and reliability was mostly valued.

We find component specific usability of the player environment to be adequate, which is in line with the found general usability. Quantitative analysis shows that students agree that component specific operability and understandability are ok (mean 3.65, in between neutral and agree). Although values for individual components differ, the differences are not significant, so we find no quantitative indications that the usability of some components is lower than for others. Qualitative analysis indicates that usability aspect operability is valued for two components, mostly valued for two components and somewhat problematic for two components, and that user error protection is somewhat problematic for one component. Software quality characteristic functionality is valued for three components, little problematic for four components, and somewhat problematic for one component, and reliability is somewhat problematic for one component. For two components we found no data.

Our quantitative evaluation may have limitations, because we used a translated version of the SUS questionnaire (in Dutch), which is not validated, and our questionnaire about component specific usability did not contain items about two components. Our qualitative evaluation may have limitations, because students could express same remarks in an online questionnaire, on paper, or oral during game sessions or afterwards in group discussions, which may have caused overvaluation of certain remarks. Our qualitative findings also do not include all usability aspects and all software quality characteristics. However, we think that students did not mention related remarks, either because they had no problems with it or it was not relevant in their context of use. For instance they will have had to deal with usability aspect 'learnability', but they made no related remarks. And characteristics maintainability and portability were not relevant in their context of use, because they do not maintain or deploy developed games.

The found usability is lower than satisfactory till very satisfactory as found in earlier superficial evaluations of the player environment (Nadolski et al., 2008; Slootmaker, Kurvers, Hummel, & Koper, 2014), despite the fact that the environment supports almost all usability heuristics and guidelines as compiled by Federoff (2002) and Ibrahim et al. (2014), and students were younger (mean age 19.3 years) than in earlier evaluations which might have some positive impact on the usability score (Lewis 2014). The lower usability might be caused by the slow internet in some institutions, but the quantitative data show no evidence for this. Our qualitative findings indicate that the cause might be that just because students were young, they expected a more realistic video game interface. It might also be that students got inadequate instruction beforehand, e.g., because one evaluated game was not yet embedded in regular education, so they missed instruction within the game. It might also be that the evaluated games themselves were, e.g., less enjoyable or challenging, or provided less support, which might negatively affect the experienced usability. Another cause might be that earlier evaluations were conducted using a different, probably simpler, player environment skin and used less components.

As a follow-up of this evaluation we plan to improve the player environment according to our findings. In a future study we will evaluate the environment again to see if our improvements result in a better usability and we hopefully are able to compare our findings with other new studies on player environments for (serious) games. Currently we are extending the player environment with new components: a dashboard component to present accumulated progress information and a history component to quickly determine what exactly has been done within a particular assignment. Also, educators who develop games expressed the need to extend the player environment with a debug window and to be able to author game content directly from within the environment, without having to use the authoring environment. And ICT developers would like to be able to more quickly build new player environment components by assembling and configuring basic interaction elements. In the near future there might be a need for a mobile or more flexible player environment skin (e.g., in screen, input element and font size) or better support for accessibility for people with disabilities.

Acknowledgements

We thank Stichting Praktijkleren for collaborating in the development and research of the games, and Wim Westera and Henk van den Brink of the Open University of the Netherlands for respectively project management and writing of the game scenarios. We also would like to thank all students and teachers of the four Dutch Regional Centers for Vocational Education (ROCs) that

participated in this study. This publication has been produced in the context of the SLEM project which has received funding from the Dutch NRO under grant agreement No. 405-14-504.

References

Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An Empirical Evaluation of the System Usability Scale. *International Journal of Human–Computer Interaction, 24*(6), 574-594.

Brooke, J. (1996). System usability scale (SUS). *Usability Evaluation in Industry*. Taylor and Francis, London.

Brooke, J. (2013). SUS: A Retrospective. *Journal of Usability Studies, 8*(2), 29-40.

Dubey, S. K., & Rana, A. (2010). Analytical Roadmap to Usability Definitions and Decompositions. *International Journal of Engineering Science and Technology, 2*(9), 4723-4729.

EMERGO (2017). *EMERGO project page*. Retrieved July 29, 2017 from http://sourceforge.net/projects/emergo/

Federoff, M. A. (2002). *Heuristics and usability guidelines for the creation and evaluation of fun in video games*. Doctoral dissertation, Indiana University.

Franzwa, C., Tang, Y., Johnson, A., & Bielefeldt, T. (2014). Balancing Fun and Learning in a Serious Game Design. *International Journal of Game-Based Learning, 4*(4), 37-57.

Gaeta, M., Loia, V., Mangione, G.R., Orciuoli, F., Ritrovato, P., & Salerno, S. (2014). A methodology and an authoring tool for creating Complex Learning Objects to support interactive storytelling. *Computers in Human Behavior, 31*, 620-637.

Gagné, R. (1985). *The Conditions of Learning and Theory of Instruction*. New York: Holt, Rinehard, and Winston.

Hamari, J., Shernoff, D. J., Rowe, E., Coller, B., Asbell-Clarke, J., & Edwards, T. (2016). Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. *Computers in Human Behavior, 54*, 170-179.

Hartson, R., & Pyla, S. (2012). *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. New York: Elsevier.

Ibrahim, A., Vela, F. L. G., Rodríguez, P. P., Sánchez, J. L. G., & Zea, N. P. (2012). Playability guidelines for educational video games: A comprehensive and integrated literature review. *International Journal of Game-Based Learning (IJGBL), 2*(4), 18-40.

ISO/IEC (2011). Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. *ISO/IEC 25010:2011*.

Klemke, R., van Rosmalen, P., Ternier, S. & Westera, W. (2015) Keep It Simple: Lowering the Barrier for Authoring Serious Games. *Simulation & Gaming, 46*(1), 40-67.

Lewis, J. R. (2014). Usability: lessons learned… and yet to be learned. *International Journal of Human-Computer Interaction, 30*(9), 663-684.

Mekler, E. D., Bopp, J. A., Tuch, A. N., & Opwis, K. (2014, April). A systematic review of quantitative studies on the enjoyment of digital entertainment games. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, 927-936. ACM.

Murray, T. (2004). Design tradeoffs in usability and power for advanced educational software authoring tools. *Educational Technology, 44*(5), 10-16.

Nadolski, R.J., Hummel, H.G.K., Van den Brink, H.J., Hoefakker, R., Slootmaker, A., Kurvers, H., & Storm, J. (2008). EMERGO: methodology and toolkit for efficient development of serious games in higher education. *Simulation & Gaming, 39*(3), 338-352.

Nadolski, R. J., Hummel, H. G. K., Slootmaker, A., & Van der Vegt, W. (2012). Architectures for Developing Multiuser, Immersive Learning Scenarios. *Simulation & Gaming, 43*(6), 825-852.

Nadolski, R. J., & Hummel, H. G. (2016). Retrospective cognitive feedback for progress monitoring in serious games. *British Journal of Educational Technology*, in press.

Nielsen, J. (1994). *Usability Engineering*. New York. Elsevier.

Sauro, J. (2011). *A practical guide to the System Usability Scale: Background, benchmarks, & best practices*. Denver, CO: Measuring Usability LLC.

Slootmaker, A., Hummel, H., & Koper, R. (2017). Evaluating the Usability of Authoring Environments for Serious Games. *Simulation & Gaming*, in press.

Slootmaker, A., Kurvers, H., Hummel, H., & Koper, R. (2014). Developing scenario-based serious games for complex cognitive skills acquisition: Design, development and evaluation of the EMERGO platform. *Journal of Universal Computer Science, 20*(4), 561-582.

Usability-First (2017). *Playability definition* [online]. Available from: http://www.usabilityfirst.com/glossary/playability/ [Accessed 10 June 2017].

Westera, W., Nadolski, R.J., Hummel, H.G.K., & Wopereis, I. (2008). Serious Games for Higher Education: a Framework for Reducing Design Complexity. *Journal of Computer Assisted Learning, 24*(5), 420-432.