

# Towards the improvement of GUARD graphical user interface

Žiga Kopušar  
Guardiaris d.o.o.  
Podjunska ulica 13, Ljubljana

+386 1 230 30 30  
ziga.kopusar@gmail.com

Franc Novak  
Jožef Stefan Institute  
Jamova cesta 39, Ljubljana

+386 1 4773 386  
franc.novak@ijs.si

## ABSTRACT

In this paper, we describe a case study of usability testing of the GUARD Control Desk graphical user interface, which is a part of the GUARD simulator and is used for exercise planning, execution and evaluation in soldier training. The usability testing was performed in the development phase of a new version of user interface.

## Categories and Subject Descriptors

H.5.2 [User Interfaces]: Graphical user interfaces (GUI), Prototyping, User-centered design

## General Terms

Design, Human Factors, Verification.

## Keywords

Usability testing, user interface, military training system

## 1. INTRODUCTION

According to [1], usability is the extent to which a system, product, or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use. The latter terms are defined in [2] as follows; effectiveness represents the accuracy and completeness with which users achieve specified goals, efficiency refers to the resources expended in relation to the accuracy and completeness with which users achieve goals, while satisfaction reflects freedom from discomfort, and positive attitudes towards the use of the product. The importance of usability has been early recognized in different aspects. While Don Norman in his famous book *The Design of Everyday Things* [3] places usability side by side with aesthetic beauty, reliability and safety, cost, and functionality, Jakob Nielsen in his earlier work [4] focusses on the design of software systems and provides general usability guidelines.

The design of complex systems such as military training simulators requires careful analysis of customers' needs and requirements in order to provide tailor-made product fulfilling their expectations. For the GUARD simulation system referred in this paper, user-centered design approach is therefore imperative. In this paper we summarize our experience and results of the usability testing of a new version of graphical interface of the GUARD Control Desk.

## 2. GUARD SIMULATION SYSTEM

The GUARD simulation system is a military training system that allows photorealistic 360° VR environments, audio and visual

effects accompanied by real-time weather and time-of-day changes. GUARD brings indoor training to the edge of real combat awareness and moves digital training borders towards real battlefield perception. The 3D real-time simulations reflect situations from the real world. Which objects take place in the 3D scene, what is the nature of the 3D scene and how the objects behave within the scene is a matter of the information recorded in the script.



Figure 1. GUARD Control Desk

The GUARD Control Desk (shown in Figure 1) is an all-in-one solution for instructors and trainees and can be added to any Guardiaris product. It is fully interoperable, interactive and features user-friendly interface for exercise planning, execution and evaluation. A large multi-touch screen and touch user interface guarantee efficient exercise planning and a best-in-class After Action Review procedure.

The simulator package includes four possible display modes of the interface. Each of them offers a different set of user controls and operations running on a specific type of device. The user interface in the editor mode, which is the subject of usability testing presented in this paper, offers the user several operations with maximum number of controllers and windows. Its task is to read, write, set and edit 3D scene. It is usually performed on developer computers with fairly strong hardware support.

### 3. USABILITY TESTING OF GUARD CONTROL DESK GUI

#### 3.1 GUI description

The library of controllers written in C++ provides means for controlling objects that are included in a given scene. Each controller carries information about object dimensions, the relative or absolute location on the screen, and about (if any) graphical icons, symbols or text with a particular meaning for the user. The user interface is used to place objects in the 3D scene. Once placed in the scene the object becomes a part of the script. All kinds of physical properties, including the basic gravity, the speed of movement, etc., are associated with an object. The interface offers integration of operations between objects, classical processes for storing, loading, cleaning the scene and operational controls ("play", "pause", "stop"). Operations for introducing objects to a scene, hiding of certain types of objects, or excluding the possibility of selecting certain types of objects can be performed via user interface.

Previous interface, has been based solely on interaction via computer mouse and keyboard. Modern technologies require completely different approaches, dealing with multi-touch displays, and other devices that are able to run graphically demanding 3D environment, but do not use conventional computer input devices. Consequently, the new user interface should support multi-touch display and additional features such as the possibility of independent setting and editing scripts via an additional software package or a dedicated application. The new concept, generated through a series of brainstorming sessions and design iterations resulted in a new user interface, which has been evaluated with the performed usability testing. The working prototype of the new user interface is shown in Figure 2.



Figure 2. Working prototype of the new GUI

When setting up the concept of usability testing, we followed the methodology presented by J. Rubin, and D. Chisnell [5] which comprises three basic test techniques: exploratory (or formative), assessment (or summative), and validation (or verification) tests at a high level. The above tests are associated with specific points in the product life cycle. In our particular case, the exploratory phase has been performed by the above mentioned brainstorming sessions and design iterations. The actual usability tests have been performed on a fully functional prototype of the new user interface and can be regarded as a combination of assessment and validation tests. The implementation of usability testing followed the guidelines presented in [6].

#### 3.2 Usability testing plan

The main goal of usability testing was to verify the adequacy of the conceptual design of the fully renewed appearance of the user interface. Consequently, the performed tests should check the ease of use, the perception of the individual sets of operations, the appropriateness of the composed sequence of operations, logic operations alone, feasibility of transformations on objects, as well as the ease of performing the actual flow of individual steps of the required test scenario. For this purpose, three step testing scenario (shown in Figures 3, 4 and 5) has been prepared.



Figure 3. First step of testing scenario

In the first step (Figure 3) the participant introduces objects in the scene. Their positions must be reasonably set into a whole. The participant thus gets acquainted with the concepts of lists of objects and groups, and with manipulator controller, which allows transformations (move, rotate, resize), and other operations (delete object, cloning facility, reset the position of the object in the initial position).

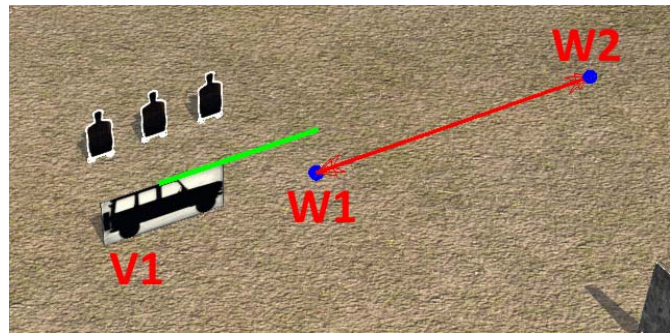
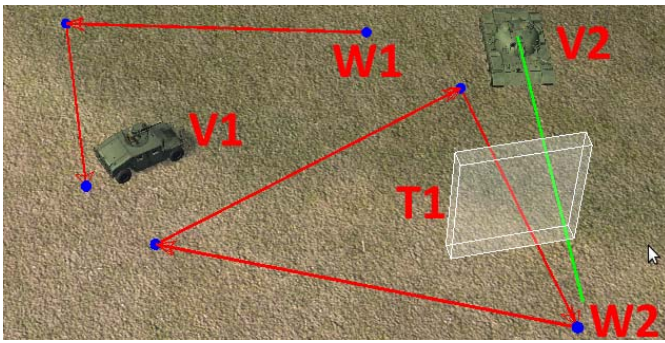


Figure 4. Second step of testing scenario

In the second step (Figure 4) the participant introduces new types of objects and becomes familiar with the operation of association between two objects. This enables integration of the "vehicle" object with the "waypoint" object, which in practice means that when the script starts, the vehicle heads towards the location of the "waypoint" object.

The main issue of the third step is the facility "trigger". The participant needs to properly connect all the objects among each other. In addition, in this step, certain attributes are assigned to the objects.



**Figure 5. Third step of testing scenario.**

Usability test plan and supporting documents were prepared following the usability test guidelines [6]. For the testing environment we used the room with a working station which is normally used for running and exercising the latest versions of software. The screen of the working station is shown in Figure 6, and the whole environment prepared for usability testing in Figure 7, respectively.



**Figure 6. The screen of the working station used for usability testing**



**Figure 7. Usability testing environment**

The selection and acquisition of participants whose background and skills are representative of those that will use the product is a crucial element of the testing process [5]. Selecting participants involves identifying and describing the relevant behavior, skills, and knowledge of the person(s) who will use your product. Within the company we managed to collect twelve participants, with different backgrounds that could be roughly categorized in three groups: a group from the hardware department, a group of participants of administrative nature, and a group originating from software industry with artists, designers

and programmers. None of them have had any previous experience with the new version of the user interface, which was the subject of the usability testing.

### 3.3 Conducting the test sessions

Testing took place in one working day and passed without major concerns or complications. Implementation of each test, on average lasted about forty-five minutes. Occasionally it was necessary to restart the editor, because it stopped working due to unexpected gestures and touches of the participants. Fortunately, there weren't many cases like this. Yet, we carefully registered any jam and placed it on the list of future urgent or less urgent corrections.

### 3.4 Usability testing results

Test results were classified in four categories:

- Opinions about appearance, suggestions on improvements.
- Utilization, logical inconsistencies of the editor.
- Quality of the editor instructions
- Programming errors and bugs in the operation of the editor or in general of the interface kernel.

About ten mistakes, opinions or suggestions for possible improvement of the appearance or functionality of the editor referred to the first and second category. Almost all participants were disturbed by imperfect control of the camera with the particular gestures. We have found that it was not the problem with gestures or users, but in the program code.

Participants' comments also justified our concern about the manipulator controller. A quarter of the participants intuitively wanted to use it in a another (and always the same) way, different than the established one. This was not a malfunction of the software code, but the problem is in a completely different presentation of an operation in a 3D scene displayed on a 2D screen.

Most of the participants did not like the automatic display of menus. They would prefer more clever automatic solution, which somehow recognizes user needs and reacts accordingly.

In the last category, about fifteen problems have been identified. Some are minor in nature, such as the improper refreshing of certain components, while others will require a more thorough investigation. In most cases in this category we deal with functional errors, or rather the requirement to change the software code at the expense of the operations of the editor.

## 4. CONCLUSION

Usability Testing results have proven to be very useful. In addition to the detected bugs, comments of the participants on the existing design and suggestions for improvements were very valuable. In the future, more effort will be given toward systematic planning of individual phases of usability testing within the complete product life cycle. We are aware that the iterative nature of usability testing requires extending the product development life cycle however with proper scheduling of testing within the design phases the benefits will be prevail. Another issue is selection of participants. A well-known fact is that one should focus its efforts on recruiting participants who are representative of the product's target users. In our case, in-house

personnel has been employed, which might have biased the results to some extent. Involving a wide range of representative users at the early stages of the development cycle is fundamental for early identification of usability problems.

## 5. REFERENCES

- [1] ISO 9241-210:2010, Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems
- [2] ISO 9241-11:1998, Ergonomic requirements for office work with visual display terminals (VDTs) — Part 11: Guidance on usability
- [3] Norman, A. D. 2002. *The Design of Everyday Things*, Basic Books; Reprint edition
- [4] Nielsen, J. 1993. *Usability engineering*, Morgan Kaufmann Academic Press
- [5] Rubin, J., Chisnell, D. 2008. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests* (Second edition)
- [6] [www.usability.gov](http://www.usability.gov)