

Analysis on Usability Heuristics for MagicHerbs Game

Selma Evelyn Cătălina Goga¹, Bianca-Cerasela-Zelia Blaga²,
Flavia Andrada Pop³, Dorian Gorgan⁴

Technical University of Cluj-Napoca

Computer Science Department

Cluj-Napoca, Romania

¹selma.goga@cs.utcluj.ro, ²zelia.blaga@cs.utcluj.ro,
³flavia.pop@student.utcluj.ro, ⁴dorian.gorgan@cs.utcluj.ro

ABSTRACT

This article discusses the evaluation stages of the MagicHerbs game using the ten usability principles of Nielsen along with the four severity ratings for the found usability problems. MagicHerbs is a new action-adventure game of type Role Playing Game (RPG) found in its prototyping stage. Three persons evaluated the application: one was the designer and developer of the game, and two other users which were already familiar with the game as they have given feedback throughout the development of the current application version. In the end, an analysis on the results of the evaluation is made and solutions for the most critical detected problems are proposed.

Author Keywords

ACM Classification Keywords

General Terms

Usability; Usability Heuristics; Game Development;

INTRODUCTION

MagicHerbs is a new action-adventure game of type Role Playing Game (RPG). The game was created by the paper's authors for educational purposes. A screenshot of the game is visible in figure 1.

The story of the game takes place in England's Middle Age, in the XVth century. The protagonist of the game is a character from the modern times, a pharmacist named Edgar, who accidentally created a magic potion which brought him back in time. His main goal is to go back home as he must find the recipe for the time travel potion. Meanwhile, he must cope with the medieval life. The best "job" he finds is that of a magician that makes healing potions for people. The story of the game is complex, however, because the game is still in the development process, its final main goal (i.e. bringing the character back to modern times) is not yet implemented.

The goal for the current implementation is to create a magic potion received from a character client from the Middle Age found inside the game. This goal can be divided into a series of tasks such as: collecting proper plants from the forest for the recipe, defending from one of the forest's predators (i.e. Wild animals or Thieves), and so on.



Figure 1. A screenshot of the MagicHerbs game. The interface which needs to be evaluated is visible.

The source code and more information about the game can be found at [9].

Since MagicHerbs game is in its prototype development stage, its usability assessment is needed.

Throughout time, authors gave their own definitions about the term "usability" for a desktop application. Primarily, usability refers to the ease of using an application. Jakob Nielsen was among the first to analyze this concept in his works [3][4][5][6][14] where he proposed ten usability heuristics used for evaluating a desktop application (see Table 1). These ten principles will be the ones used for evaluating the interface of our game.

This paper is structured as follows. In the next section, we give an outline of related works (RELATED WORK section), followed by a detailed description of the usability metrics used to evaluate MagicHerbs (USABILITY METRICS section). Next, the process of evaluating the video game along with the results are presented (EVALUATING GAME'S USABILITY section). The results from the evaluation stage are analyzed in RESULT ANALYSIS section, and possible solutions for improvement are discussed.

RELATED WORK

Nielsen's principles are still used worldwide since their creation for many computer programs in which the interaction between the user and the application needs to be evaluated. However, Nielsen's heuristics do not evaluate specific features of specific applications [15]. For these

cases, mappings of these principles to applications from different domains have also been proposed [1][12][13].

In [1] the authors noticed that a video game cannot be considered a simple desktop application. They defined the usability of a video game as being “the degree to which a player is able to learn, control, and understand a game”. Video games are a combination of art and science [2]. Because of this, new usability heuristics have been proposed especially for video games throughout time [1][8][10][11].

The authors in [8] are one of the firsts to study the need of creating new usability principles for video games. They have questioned whether the ten usability heuristics of Nielsen can still be used for analyzing the interaction between a player and the game. The conclusion they reached was that as long as one wants to evaluate only the interface of the game, Nielsen’s principles still stand. However, for analyzing the playability of a game, they are not sufficient, and other principles are proposed for this. This is because the purpose of a desktop application is using the application, temporary, in order to manage an external need, while, a video game’s purpose is using (playing) it for unlimited period in order to obtain an internal satisfaction.

Following the works of [8], authors in [7] agreed that a complete game analysis should contain also the playability.

Many studies have been made on analyzing the playability. In articles [2] and [10] new playability and usability heuristics have been proposed. Authors in [2] proposed PLAY, an iterative manner for evaluating a game. A series of categories have been proposed, each containing several heuristics for evaluation. Compared with [1], the authors in PLAY [2] try to address every aspect of a game when evaluating it, the interface, the ease of interacting with the game, the users cognitive experience and so on. Same ideas are presented in [10].

Although a complete game analysis contains evaluation of usability and of the playability, MagicHerbs is in its development stage and only a prototype is available. In this prototype only a few scenarios are presented, compared to the complexity of the final game. For now, analyzing only the interaction between a player and the application through the interface is wanted.

Notice that at the basis of any newer usability heuristics defined especially for games, lies the set of Nielsen’s ten principles of usability [5]. Recall that authors in [8] stated that for analyzing the interface of a game, these ten principles are enough. Following these observations, the article will evaluate the usability of the game MagicHerbs using the principles defined by Nielsen. The four severity score concept [4] will also be used to evaluate the severity of a problem which violates one of the usability principles.

USABILITY METRICS

Two usability evaluation concepts proposed by Nielsen Jakob will be used to evaluate MagicHerbs prototype:

- A. The ten usability heuristics [5]
- B. The Severity Scale [4]

A. The ten usability heuristics

These ten usability heuristics were used as guideline for evaluating the interaction between a user with the MagicHerbs game through the proposed interface. The principles can be seen in table 1.

The ten usability heuristics [5]	
H1	Visibility of system status
H2	Match between system and the real world
H3	User control and freedom
H4	Consistency and standards
H5	Error prevention
H6	Recognition rather than recall
H7	Flexibility and efficiency of use
H8	Aesthetic and minimalist design
H9	Help users recognize, diagnose, and recover from errors
H10	Help and documentation

Table 1. The usability principles used for MagicHerb video game.

H1 checks whether the feedback received from the system at the moment of interaction with it alright. Ideally, the feedback should be received in a reasonable time. H2 follows how familiar is for the user its interaction with the system. This interaction should be user oriented, rather than system-oriented. H3 checks if an unwanted action which the user made in the system can be undone and easily redone. H4 follows the consistency of the game, which mean that the user should not wonder whether different words, situations or actions mean the same thing. H5 and H9 are concerned with the errors the system might produce. The system should try to prevent errors. If, however, an error occurs, the user should be helped in order to identify and recover from it. H6 concerns with minimizing the user’s memory load (e.g. the user should not need to remember information from one dialog in order to understand information from another dialog). H7 refers to the ease an inexperienced or an experienced user interacts with the system. H8 is concerned with the aesthetics part of the interface (e.g. are the colors chosen pleasant for the user’s eye, is the text on the screen easily readable, and so on). Finally, H10 focuses on providing an accessible help information from the user when it needs one.

For the game MagicHerbs, the evaluators had to grade with a score from 1 to 10 on how good these principles are applied to each of the six scenarios (tables 3-9) implemented in the current version of the game.

B. The Severity Scale

Another concept which helps evaluating the usability of an application and proposing solutions for it is the Severity Scale also proposed by J. Nielsen [4]. These scale is popular for usage when evaluating a video game.

The idea behind them is that for each problem that violates one of the ten heuristics principles, a rating from 1 to 4 is assigned to it which illustrates how severe that problem is (Table 2). In Table 2, SR stands for Severity Rating.

The Severity Scale[4]	
SR 1	Cosmetic problem
SR 2	Minor problem
SR 3	Major problem
SR 4	Usability Catastrophe

Table 2. The four severity ratings assigned to each problem found for the interface of the game.

The evaluators of the game will have to explain the problem that violates a heuristic, found problem when giving a grade for a heuristic lower than 10. For the described problem they also have to assign a severity rating which they consider appropriate for it.

EVALUATING GAME'S USABILITY

MagicHerbs has been evaluated by three people. The first evaluator was the designer and developer of the game, called Designer, and the second evaluation has been done by two evaluators generically called Evaluator1 and Evaluator2.

At first, the cognitive evaluation of the game has been made by the designer of the game following Nielsen's usability principles and severity scale. Each scenario of the six (see tables 3-8) implemented for the prototype of the game was evaluated. The problems which the designer considered to be severe were solved before giving the game for evaluation to the evaluators. The major problems solved were concerned with H7, H8 and H10.

For H8, it has been noticed that a series of texts which were printed on the screen with white color were not very visible because of the light colored background. The solution was to outline the contour of the text with black color, which made the text more legible. Other printed texts had a font which was too small.

For H10, some supplementary messages were added in order to help the user understand the goal of the current game process. A HELP button which showed the controls of the game was missing so it was added.

It has also been noticed that the movement of the mouse to navigate the gameplay interface was hard because the mouse movement was also tied with the movement of the main camera of the game. This violated H8. To solve this a new keyboard key event was added: once the user would press that key, the main camera would freeze and the user would have the freedom of moving the mouse freely on the screen with ease.

After the initial evaluation made by the designer of the game and after the major problems found were solved, the game was passed on to the two evaluators.

The two users of the game which evaluated its usability were familiar with the concept of the game from the start as they have given feedback throughout the game's design process. One user had a high experience with PC video games (Evaluator1), while the other had a high experience with mobile phone games, but not with ones for PC (Evaluator2). Also, both evaluators had high experience with the computer as both graduated Computer Science BS.

For the evaluation of the game's usability by the evaluators, two stages were followed:

- A. **The individual evaluation stage** – in this stage, each user evaluated the game independently
- B. **The group evaluation stage** – in this stage, users presented their evaluation to one another and discussed the problems found by each.

A. The individual evaluation stage

Initially, the two evaluators received the six scenarios for MagicHerbs which they were asked to test before starting the evaluation. The six scenarios have been chosen based on the interactions needed between the user and the game in order for the users to give a better evaluation of the game's interface. The six scenarios can be viewed in the Tables 3-8. After the users played at least once all the scenarios of the game, they received a document with several tables which they had to fill in. There were six tables for each gameplay scenario, each with ten rows, each for a usability heuristic.

At first, each one of Nielsen's usability heuristic [5] was properly explained to the user's giving them examples about what to look at in the game's interface for each. For each scenario the ten principles of usability had to be rated with a score from 1 to 10. A lower grade meant that the heuristic for that scenario was violated while a higher grade meant the opposite.

If a user graded one of the principles for a scenario with a score lower than 10 they had to also describe the problem and rate it with a severity score from 1 to 4 based on Nielsen's severity scale [4]. It has been explained to them what each rating means.

The evaluation tables filled by the users along with their observations for each violated heuristic can be seen in

tables 9 to 14. For each stated problems at the end an R and a letter appear. That represents the severity rating score for that issue.

S1: Moving on the game's map	
Task 1	Move front, right and left using the keyboard arrows
Task 2	Press "0" key while pressing up arrow in order to run
Task 3	Use the mouse movements to move the view point of the camera up and down
Task 4	Scroll mouse wheel for zoom in and out.

Table 3. First scenario to evaluate in MagicHerbs game.

S2: Picking up a plant	
Task 1	Press one of the CTRL keys to block the screen.
Task 2	While still pressing CTRL, Hover mouse over a plant
Task 3	Release CTRL and move closer to the plant
Task 4	Mouse right-click onto the screen to pick plant

Table 4. Second scenario to evaluate in MagicHerbs game.

S3: Attacking a forest predator	
Task 1	Mouse left-click to attack when close to a predator.
Task 2	Low health in health bar? Run from the predator until it stops following Ellyn.
Task 3	Release all keys and wait until health bar fills.

Table 5. Third scenario to evaluate in MagicHerbs game.

S4: Winning the game	
Task 1	Check recipe plants quantity in panel "View Order" vs. plants quantity in inventory.
Task 2	Left-click Send Order button
Task 3	Recipe plants quantity is equal to inventory plans quantity? A message "You WON!!!" will appear in the center of the screen and the entire game will freeze.
Task 4	Recipe plants quantity is not equal to inventory plans quantity? A message which contains game tips on how to win

	the MagicHerbs game is displayed.
--	-----------------------------------

Table 6. Forth scenario to evaluate in MagicHerbs game.

S5: Visualizing additional information - Help	
Task 1	Left Click HELP button.
Task 2	Left Click again HELP and notice the change.
Task 3	Left Click Potion Order button.
Task 4	Left Click again Potion Order button and notice the change.

Table 7. Fifth scenario to evaluate in MagicHerbs game.

S6: Loosing the game and exiting/restarting it	
Task 1	Health bar is empty? A message "You LOSE!!" appears instantly in the center of the screen.
Task 2	Left click on "X" sign to exit the game.
Task 3	In order to restart game exit first the game and then reopen the application.

Table 8. Sixth scenario to evaluate in MagicHerbs game.

B. The group evaluation stage

After the individual evaluation, the two evaluators were asked to discuss the evaluation results with one another. Some problems were identified by only one evaluator and in this stage the other evaluator sometimes agreed to the problems found by the other and scored them with a severity rating accordingly. The problems which were identified by only one evaluator and the other agreed to him are also visible in the tables from 9 to 14. For those problems, their statement is written in the same manner for both evaluators. For example, in table 9, both evaluators found H2 violated because "The controls are hard to use. I would have preferred the keyboard keys "A", "D", "W", "S" instead of the arrows."

In tables 9-14, (SRx) stands for Severity Rating x, where x is an integer number from 1 to 4, as described for Table 2.

S1	Usability Evaluation			
H1	10		9	Avatar doesn't move left right without going forward(SR1)
H2	5	Non-intuitive controls. Better use a,d,w,s for movement. (SR4)	5	Non- intuitive controls. Better use a,d,w,s for movement. (SR4)
H3	7	The movement keys don't	8	Character moves to fast

		always do what is expected. (SR2)		when trying to change direction (SR3)
H4	9	Left right movement not consistent. (SR2)	10	
H5	9	User should be aware that he can't move back-wards! Problem with object collisions (SR2)	10	
H6	10		10	
H7	7	System not flexible because of controls used(SR2)	8	System not flexible because of controls used(SR3)
H8	10		10	
H9	8	No error message appears when getting stuck in an object (SR1)	9	A map of the environment is missing.(SR4)
H10	8	Help not enough. Lack of backward movement control not stated. (SR1)	9	Help insufficient for current controls. (SR1)

Table 9. Evaluation of first scenario from the two users. First column: numbered heuristics, next two columns: Evaluator1's evaluation, last two columns: Evaluator2's evaluation.

S2	Usability Evaluation			
H1	10		10	
H2	10		10	
H3	8	Can't undo after collecting a plant. (SR2)	9	Hard to get close enough to a plant (SR2)
H4	9	Two same plant types have different names (SR1)	10	
H5	10		10	
H6	10		10	
H7	9	User shouldn't get so close in order to collect	8	User shouldn't get so close in order to collect

		a plant.(SR2)		a plant.(SR2)
H8	10		10	
H9	9	Plant is incremented with 2 instead of one when collecting it. User can't undo. (SR1)	10	
H10	10		10	

Table 10. Evaluation of second scenario by the two users. First column: numbered heuristics, next two columns: Evaluator1's evaluation, last two columns: Evaluator2's evaluation.

S3	Usability Evaluation			
H1	7	Sometimes enemies disappear from scene (SR3)	9	Don't know how close to get to attack a predator. Give feedback-Display a message!(SR1)
H2	10		10	
H3	8	Bug when sometimes trying to run from predator-life bar empties instantly. (SR1)	10	
H4	8	If two enemies attacking, player doesn't know which is affected by attack. (SR3)	10	
H5	5	Error prevention messages are necessary!(SR2)	7	Error prevention messages are necessary!(SR1)
H6	10		10	
H7	9	Hard to run from predator.(SR2)	8	Hard to run from predator.(SR3)
H8	10		10	
H9	8	See H5 and H3 (SR3)	9	See H5 and H3 (SR2)
H10	8	Lack of power damage information.	10	

		(SR2)		
--	--	-------	--	--

Table 11. Evaluation of third scenario by the two users. First column: numbered heuristics, next two columns: Evaluator1's evaluation, last two columns: Evaluator2's evaluation.

S4	Usability Evaluation			
H1	10		10	
H2	10		10	
H3	10		8	Hard to follow recipe.(SR1)
H4	8	Can't recognize plants well because some names are written bad (SR1)	10	
H5	9	Player must always read the order recipe in order to know quantities.(SR 1)	10	
H6	5	Hard to follow recipe. Must memorize quantities. Use something else instead of pure text. E.g. a bar which fills while you pick up a plant. (SR3)	7	A map with where to find a certain plant is needed(SR4)
H7	8	Hard to move mouse cursor to buttons. Unintuitive CTRL key.(SR2)	6	Hard to move mouse cursor to buttons. Unintuitive CTRL key.(SR3)
H8	10		10	
H9	7	User needs a log in which his recent actions are mentioned. (SR3)	10	
H10	9.5	Goal of the game unclear at first.	9	Not clear at first how to win game. (SR3)

		(SR1)		
--	--	-------	--	--

Table 12. Evaluation of fourth scenario by the two users. First column: numbered heuristics, next two columns: Evaluator1's evaluation, last two columns: Evaluator2's evaluation.

S5	Usability Evaluation			
H1	10		10	
H2	10		10	
H3	10		10	
H4	10		10	
H5	10		10	
H6	10		10	
H7	10		10	
H8	9	Help menu text is hard to read (SR1)	9	View Order button doesn't appear like a button + text to small(SR1)
H9	10		10	
H10	10		10	

Table 13. Evaluation of fifth scenario by the two users. First column: numbered heuristics, next two columns: Evaluator1's evaluation, last two columns: Evaluator2's evaluation.

S6	Usability Evaluation			
H1	8	System exits too quickly. "are you sure" dialog expected. (SR3)	10	
H2	10		10	
H3	10		9	Lack of restart button. Lack of pause button.(SR2)
H4	10		10	
H5	10		10	
H6	10		10	
H7	10		10	
H8	10		10	
H9	8	Can't undo after exit button pressed.(SR1)	10	
H10	10		10	

Table 14. Evaluation of sixth scenario by the two users. First column: numbered heuristics, next two columns: Evaluator1's evaluation, last two columns: Evaluator2's evaluation.

RESULT ANALYSIS

All the discovered problems from all the stages of the usability evaluation have been analyzed. A mean of the grades received for each heuristic in each scenario has been computed. Figure 3 illustrates these values.

The problems have been counted for each of the four severity ratings. Notice that sometimes the evaluators noticed the same problem but they gave different scores for each. In order to cope with this issue, we count the total number of unique problems identified only. For the ones where the severity score is different, we will take into consideration both scores. Adding these values will give the final number of system problems.

The total number of unique problems that violated usability heuristics found in all scenarios is **38** (almost 19 problems found by each evaluator), from which:

- 16 had a **severity rating 1**
- 14 had a **severity rating 2**
- 11 had a **severity rating 3**
- 4 had a **severity rating 4**

Note that the system has actually **45** problems in total noticed but from them only **38** are unique.

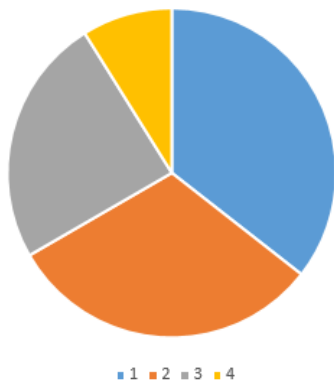


Figure 2. The ratio between the number of problems detected for each severity score.

Pie chart from figure 2 illustrates better the ratio between the number of problems detected for each severity score.

Part of the problems that both users agreed on were:

- *“Non-intuitive controls. Better use a,d,w,s keys for movement.” Scenario 1- H2 problem - Rating 4*

- *“User shouldn't get so close in order to collect a plant. System not flexible.” Scenario 2- H7 problem - Rating 2*
- *“More error prevention messages are necessary! For predator disappearance and sudden reappearance error” Scenario 3 – H5 problem – Rating 2 and 1*

Although the principles of usability are clearly defined and both evaluators tested the same scenarios, a difference in how evaluators perceive the same problem can be seen. In Scenario 4, H5, both evaluators agreed on a problem but the difference is that one gave a severity rating of 2 to it and the other a severity rating of 1.

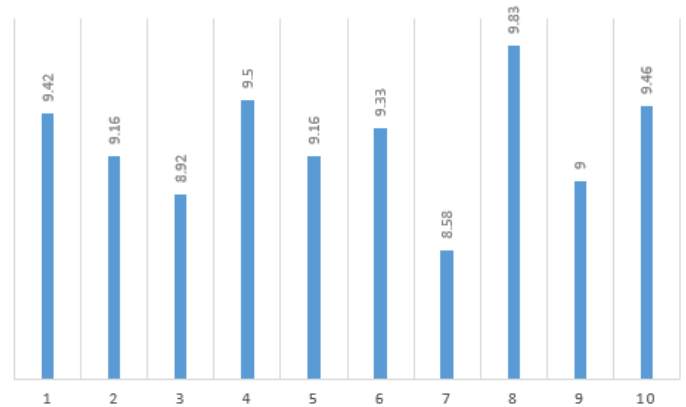


Figure 3. The mean of the grades obtained for each of the ten heuristics in all six scenarios of the game. On the horizontal axis we have noted the ten principles numbered and above each bar we have the real value of the mean to which the bar corresponds.

The method for solving the problems found by the evaluators is done by first noticing the heuristics with the highest severity score or with the lowest mean grade obtained. These correspond with the most critical points from the system. We will discuss solutions only for the critical points of the system in this article. **Table 15** shows a summary of the final results.

We observe there are heuristics which received a good final grade in the end but their severity score is the highest (e.g. H2-with SR average 4; H6 – with SR average 3.5). For these two heuristics we will consider them the most critical problems from the current game version. From the feedback given by the evaluators we observe that these problems appear because of the proposed keyboard keys for movement, the arrow keys. These keys are not popular for the movement of a player in a RPG type game. Instead, the evaluators proposed “a,d,w,s keys”, the most popular keys used in game industry for this type of games. This will be implemented in a newer version of MagicHerbs in order to ease the interaction of the player with the game world. For

H6, the reviewers mentioned that it is best to find another graphical way of giving information about the needed quantities of the recipe (instead of text) to the player and to give a map with possibilities where a type of plant might grow. The solution found by the player which implied using a bar just like the health bar which fills gradually while one tries to collect all plants needed for recipe appears to be the best.

Nielsen J. Heuristic No.	Heuristic grade average	No. problems found	SR average
H1	9.42	4	2
H2	9.16	2	4
H3	8.92	7	1.85
H4	9.5	4	1.75
H5	9.16	4	1.5
H6	9.33	2	3.5
H7	8.58	8	2.37
H8	9.83	2	1
H9	9	7	2.28
H10	9.46	5	1.6

Table 15. Summary of the results. First column represents the ten usability heuristics, second column the average grade received for the six scenarios. Third column shows the number of problems detected for all six scenarios for that heuristic and the last column the computed mean severity rating (SR) for those problems is shown.

CONCLUSIONS

MagicHerbs is a new video game project found in its prototype development stage.

A series of heuristic evaluations have been made by the developer of the game at first and then by two reviewers which were referred to as users/evaluators of the game throughout the article. The evaluators were familiar with the system as they gave feedback starting from the early development stages of the game. As a guideline, the heuristic evaluation used the ten usability principles and the Severity Scale of J. Nielsen [5]. The discovered problems were analyzed and solutions for the most critical ones have been proposed.

REFERENCES

[1] Pinelle, David, Nelson Wong, and Tadeusz Stach. "Heuristic evaluation for games: usability principles for video game design." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2008.

[2] Desurvire, Heather, and Charlotte Wiberg. "Game usability heuristics (PLAY) for evaluating and designing better games: The next iteration." International Conference

on Online Communities and Social Computing. Springer, Berlin, Heidelberg, 2009.

[3] Nielsen, J. How to Conduct a Heuristic Evaluation. www.useit.com/papers/heuristic/heuristic_evaluation.html, 2007.

[4] Nielsen, J. Severity Ratings for Usability Problems. <http://www.useit.com/papers/heuristic/severityrating.html>, 2007

[5] Nielsen, J. Enhancing the Explanatory Power of Usability Heuristics. In Proc. CHI 1994, 152-158.

[6] Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, NY, 1994.

[7] Desurvire, H., Caplan, M., and Toth, J. A. Using heuristics to evaluate the playability of games. Ext. Abstracts CHI 2004, 1509-1512.

[8] Federoff, M. Heuristics and Usability Guidelines for the Creation and Evaluation of Fun in Video Games. Indiana University Master of Science Thesis, 2002.

[9] MagicHerbs_Game: <https://github.com/selmagoga/MagicHerbs>

[10] Febretti, Alessandro, and Franca Garzotto. "Usability, playability, and long-term engagement in computer games." CHI'09 Extended Abstracts on Human Factors in Computing Systems. ACM, 2009.

[11] Jørgensen, A. H., Marrying HCI/Usability and computer games: a preliminary look. In Proc. of NordiCHI, (2004), 393-396.

[12] Thitichaimongkhon, Kritpapon, and Twittie Senivongse. "Enhancing usability heuristics for android applications on mobile devices." Proceedings of the world congress on engineering and computer science. Vol. 1. 2016.

[13] Rusu Cristian, et al. "Usability heuristics for grid computing applications." Proceedings ACHI (2011): 53-58.

[14] Nielsen, Jakob. Usability engineering. Elsevier, 1994.

[15] Quiñones, Daniela, Cristian Rusu. "How to develop usability heuristics: A systematic literature review." Computer Standards & Interfaces 53 (2017): 89-122.