Computers in Human Behavior 54 (2016) 124–133

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Full length article

Reduced GUI for an interactive geometry software: Does it affect students' performance?

Simone S. Borges [a], Helena Macedo Reis [a], Leonardo B. Marques [a], Vinicius H.S. Durelli [b], Ig Ibert Bittencourt [c], Patrícia A. Jaques [d], Seiji Isotani [a, *]

[a] *University of São Paulo, São Carlos, Brazil*
[b] *University of Groningen, Groningen, The Netherlands*
[c] *Federal University of Alagoas, Maceió, Brazil*
[d] *University of Vale do Rio Sinos, Rio Grande do Sul, Brazil*

A R T I C L E   I N F O

Article history:
Received 19 January 2015
Received in revised form
29 July 2015
Accepted 30 July 2015
Available online 14 August 2015

Keywords:
Interactive geometry software
iGeom
Graphical user interface
Experimental study
Interactive learning environment

A B S T R A C T

Purpose: The purpose of this paper is to describe an experimental study to reduce cognitive load and enhance usability for interactive geometry software.
Design/methodology/approach: The Graphical User Interface is the main mechanism of communication between user and system features. Educational software interfaces should provide useful features to assist learners without generate extra cognitive load. In this context, this research aims at analyzing a reduced and a complete interface of interactive geometry software, and verifies the educational benefits they provide. We investigated whether a reduced interface makes few cognitive demands of users in comparison to a complete interface. To this end, we designed the interfaces and carried out an experiment involving 69 undergraduate students.
Findings: The experimental results indicate that an interface that hides advanced and extraneous features helps novice users to perform slightly better than novice users using a complete interface. After receiving proper training, however, a complete interface makes users more productive than a reduced interface.
Originality/value: In educational software, successful user interface designs minimize the cognitive load on users; thereby users can direct their efforts to maximizing their understanding of the educational concepts being presented.

1. Introduction

The interplay between computer systems and human beings occurs through a graphical user interface (GUI). Improperly designed GUIs (i.e., interfaces that do not meet usability criteria) often hinder how users interact and access the underlying functionality (Nielsen, 1993). As a result, users might end up performing wrong operations, thereby reducing their productivity and even tampering with parts of the system. GUIs developed according to usability patterns provide a number of benefits to their users: they (i) optimize users' productivity; (ii) help users to quickly memorize the available functionalities; and (iii) mitigate interaction problems.

In the context of educational software for geometry, the development of interfaces can influence how learners explore and understand the concepts shown on the computer screen (Sedig & Liang, 2006). Interactive Geometry (IG) software is computer programs tailored toward geometry education. IG software allows learners to interact with geometry objects and dynamically construct their knowledge (Isotani & Brandão, 2008). Recent studies suggest that IG software containing interfaces that show a great number of graphical elements and features are not adequate for beginners (Kortenkamp & Dohrmann, 2010; Schimpf & Spannagel, 2011). According to these studies, when learners use GUIs containing a large number of functions, they spend a significant amount of time “trying to find certain features” instead of learning the subject (e.g. Mathematics).

* Corresponding author.
E-mail addresses: simone.borges@gmail.com (S.S. Borges), helenamcd@gmail.com (H.M. Reis), leobmarques@gmail.com (L.B. Marques), vinicius.durelli@gmail.com (V.H.S. Durelli), ig.ibert@gmail.com (I.I. Bittencourt), patricia.jaques@gmail.com (P.A. Jaques), sisotani@icmc.usp.br (S. Isotani).

http://dx.doi.org/10.1016/j.chb.2015.07.064
0747-5632/

Although some studies have discussed the development of interfaces for IG software, little effort has been made to carry out empirical studies to evaluate how GUIs can be used to avoid problems in the learning processes (Mackrell, 2011). There are approaches to cope with the complexity found in GUIs. These approaches turn complex GUIs into more user-friendly ones. For instance, one approach consists of enhancing adaptability by allowing the user to either hide or disable extraneous features. By keeping only the graphical elements that are used more often, users tend to be more productive (Schimpf & Spannagel, 2011). However, the existing approaches are not specific to IG software (Kortenkamp & Dohrmann, 2010).

Designing effective GUIs for IG software is important because GUIs play a pivotal role in the learning process. Learning consists of transferring information from working memory to long-term memory (Coyne, Baldwin, Cole, Sibley, & Roberts, 2009). Nevertheless, our working memory can hold a limited amount of cognitive load, which means that when faced with difficult tasks we should use our working memory effectively. Within this context, learners that study geometry through IG software with low usability waste most of their cognitive load in learning how to interact with the underlying IG software. Thus, GUIs for IG software should be friendly enough not to exceed users' working memory capacity.

This study aims at understanding how the interfaces of IG software affect the learning process and the productivity of their users. In order to investigate this topic, an experiment involving 69 undergraduate students was carried out using the IG software called iGeom (Isotani & Brandão, 2008). Most subjects involved in the experiment stated they had advanced computer skills and a working knowledge of geometry. Only four students had prior experience with IG software, and only one student had prior experience with the particular IG software used in the experiment.

In the next sections, we present background, our experiment, and the conclusions of this study. The concepts of IG software, cognitive load, and usability are described in Section 2. Section 3 presents related work and similar experiments already carried out in the area. Section 4 describes the experiment we carried out using two versions of interface for iGeom, namely, **complete** and **reduced** interfaces. Section 4 also discusses the results of our experiment. Section 5 suggests future work and Section 6 presents concluding remarks.

2. Background

Interactive geometry software (IGS; dynamic geometry environments, DGEs; or dynamic geometry systems, DGSs) is computer programs developed with the goal of enabling students to explore geometry concepts through dynamic manipulation of geometric objects (e.g., Lines, circles and dots) (Erez & Yerushalmy, 2006; Isotani & Brandão, 2008; Roanes-Lozano, 2003). IG software implements the conventional tools employed in classroom settings to teach geometry, such as a ruler and compass using computational resources. The term “geometry” refers to the branch of mathematics that studies properties and relations of geometric objects. In this context, IG software allows students to create abstract representations of geometric objects and concepts to measure and manipulate them. These activities allow students to receive quick feedback after handling an object on screen. Consequently, students can test conjectures and hypotheses and find new relationships and properties based on a constructivist approach (Hollebrands, 2003).

During the learning process, students interacting with the software are able to visualize geometric constructions via GUI. Furthermore, they are able to interact with the features of the software and easily understand the information through these visualizations (Shimomura, Havannber, & Hafsteinsson, 2013). However, Baker, Greenberg, and Gutwin (2001) and Laborde (2007) suggest that the development of IG software should take into consideration not only pedagogical aspects, but also the design of the interface. The reason is to avoid developing software that tends to support a superficial sort of teaching, causing frustration among the students, who may struggle using the software and ultimately not direct their attention to the task that really matters: to learn geometry (Kortenkamp & Dohrmann, 2010; Schimpf & Spannagel, 2011).

Previous studies have shown that students who use IG software to learn geometry are more committed than students learning with traditional tools, such as rulers and compasses (Erbas & Yenmez, 2011). Research findings indicate that IG software encourages students to develop their own hypotheses and find new ways to solve the proposed problems (Isotani & Brandão, 2008). Yet, during the learning process students need to learn at the same time, both mathematical concepts and how to use the IG software (e.g., understand the interface and functions of the software).

2.1. Cognitive load theory and usability

Schimpf and Spannagel (2011) have shown that one of the main difficulties reported by students while learning how to use IG software is related to the wide variety of features in their interfaces. A large number of features can lead to ambiguity, confuse the students with too many details, and cause frustration and demotivate students. In other words, GUIs of IG software containing too many graphical elements may hinder the learning process, requiring users to experience high cognitive loads. As pointed out by Sedig and Liang (2006), the cognitive abilities of students are limited and should be directed to help them understand mathematical concepts and not be wasted learning how to use the software interface.

The cognitive load refers to the demands placed on the working memory of learners during the learning process. This concept is the basis of the cognitive load theory (CLT) (Paas, Renkl, & Sweller, 2003). According to this theory, humans have limited information-processing capacity. Because of that, we (humans) have difficulty memorizing past concepts for a long period during instruction. Cognitive load theory is concerned with how to maximize the performance of students investigating ways in which statements/interfaces must be presented and the type of activities in which learners must engage. The knowledge of human cognitive architecture provides the foundation that underlies the area, especially memory and long-term memory working.

CLT classifies cognitive load into three different types: intrinsic, germane and extraneous cognitive load (Sweller, Ayres, & Kalyuga, 2011). The ***intrinsic cognitive load*** is related to the complexity of a given task that must be processed by the learner and cannot be modified by an instructor. ***Germane cognitive load*** is the load utilized to the processing, construction and automation of schemas. The ***extraneous cognitive load*** is generated by the way in which information is shown to learners. It means that extraneous cognitive load demands mental efforts that do not promote learning and can be attributed to the design of the instructional materials (Chandler & Sweller, 1991).

The interface designers are concerned with the extraneous cognitive load, such as the level of irrelevant information presented to users. As shown in Fig. 1, if the interface presents irrelevant elements, learners need to figure out what information is important and what information is not relevant to their learning. Unlike intrinsic load, that is constant (considering the task to be performed); the other loads can be controlled (Gog & Paas, 2012). Therefore, to design an educational system that helps students to

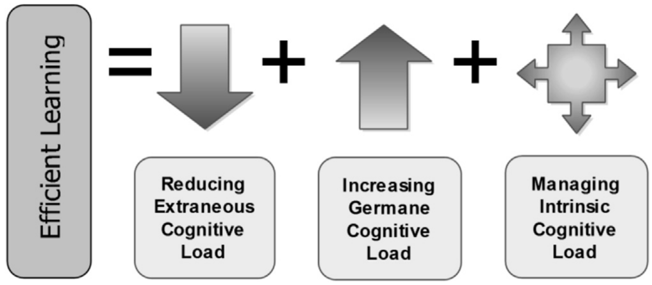

Fig. 1. Balancing three sources of cognitive load to maximize efficiency (adapted from Sweller et al., 2011).

understand the content and construct their knowledge more effectively, it is important to create GUIs that *manage intrinsic cognitive load* efficiently, *reduce the extraneous cognitive load* and *maximize germane cognitive load* (Plass, Chun, Mayer, & Leutner, 2003).

In this context, CLT influences how to provide better usability to educational software systems. In human–computer interaction parlance, usability refers to the simplicity and ease with which a GUI can be used (Tidwell, 2011). The interface is the point of interaction between humans and an abstract or physical object. Educational software systems need to be built using standards of quality and usability so that they can be properly used and help users (i.e., students) to achieve their goals (i.e., learning). Usability can be improved by making learning easier, reducing the time spent on memorizing operations, and pruning away interaction errors.

Interfaces that have more features, although useful for experienced users, can be problematic for novice users. When searching a certain feature, an inexperienced user may take longer to find it when it is somewhat hidden by other features. This indicates that the interface has a high load and it takes an extraneous amount of working memory for the user to learn it.

However, despite the importance of educational software interfaces, the various forms of interaction available with the IG software interfaces have not been widely explored. There are few studies about exploiting the diversity of interactions and the devices that they run on (Reis, Borges, Griffiths, Moro, & Isotani, 2013).

3. Related work

Oliveiras and Santos (2003) devised a set of tasks to simplify the realization of heuristic analyses in the context of IG software. These analyses are important to detect problems in IG's interfaces. Sedig and Liang (2006) developed similar research to identify factors that affect the learning and the cognitive processes during a mathematics learning process using interactive software. As a result, they created a framework and a group of 12 interactive factors that can help evaluate math-teaching software. Mackrell (2011) identified important points of conflict not only between the functionality and the complexity, but also between stativity and dynamicity of the interface. Both characteristics have the power to affect learning using IG software. However, the results from Mackrell's study do not offer strategies that can be easily applied to design interfaces. Improving IG software interfaces is desired because it can boost productivity and make the learning of fundamental mathematics abstractions and concepts easier. Currently, there is no agreement in the community about how to project and implement an interface for IG software.

To provide more insight on the impact of IG interface on users, Schimpf and Spannagel (2011) carried out an experiment to investigate the level of usability of the interface of the IG software called Cinderella (Richter-Gebert, Kortenkamp, & Ulrich, 2012). The main objective of their experiment was to measure the amount of time a student spent finding an icon in the interface in question. After analyzing the results, the researchers discovered that it would not be necessary to reduce the amount of icons of an interface to elevate its usability. This suggests that, when using IG software functionalities, the amount of cognitive load experienced is somewhat independent of the number of icons available on the interface. However, the research indicates that other facts may have influenced the result of the experiment. For example, the authors did not check the level of learning of the students. Thus, new experiments should be carried out to identify the factors that can affect the learning and the use of IG software.

4. A reduced interface for interactive geometry software: igeom

iGeom is a multi-platform IG software developed at the Institute of Mathematics and Statistics at the University of São Paulo (Isotani & Brandão, 2008). Similar to other IG programs, iGeom allows users to perform IG basic operations such as the creation of geometric objects (e.g. points, segments, lines, circumferences, angles, and so on) and the manipulations of their characteristics (e.g. change in color, size, name, etc). It is also possible to perform complex operations such as development of fractals, scripts, and functions that can be applied in geometric constructions and objects.

In previous research with the use of iGeom (Isotani & Brandão, 2008; Isotani, Mizoguchi, Inaba, & Ikeda, 2010), we observed a few interesting contradictory results. On the one hand, several learning benefits were found when students have frequently used the software. On the other hand, novice students have struggle to learn and deal with the software's GUI, in particular due to the large number of icons/buttons available. Thus, to have a better understanding on how a reduction in the number of icons/buttons affect students' interaction with GI software became an important questions to be answered.

Our initial assumption was that a reduced interface (i.e. with less icons/buttons) could be the best solution. Nevertheless, as discussed by Findlater and McGrenere (2007) a user interacting with a simplified GUI may not learn more than a user interacting with a GUI with more functions and features. In addition, Kortenkamp and Dohrmann (2010) remarks that, based on their experience, even if a GUI is considered more appropriate in relation to another, usually, users will tend to prefer the interface presented to them first. Although these findings are important, they are not empirically evaluated. Furthermore, they are contradictory with others findings discussed in section 3.

Therefore, to contribute to better understand the impact of GUI of GI software on users, we developed two versions of the iGeom's GUI, namely, complete and reduced interfaces. The first one, i.e. **complete interface**, *incorporates as many features as possible in the toolbar*, exposing the users to the full functionalities to build geometric objects of iGeom. The second one, i.e. **reduced interface**, *displays only a basic subset of features in the toolbar* by grouping similar functionalities. Fig. 2 shows the complete version of iGeom's interface (left) and the reduced one (right). The menu bar that presents functionalities not related to geometric construction (such as, save file, open file, delete, and so on) are the same in both interfaces.

Two criteria were used to group the functionalities in the reduced interface: (i) math-based criteria and (ii) interaction-based criteria. According to the first one, all math tools were grouped according to their similarity. For example, buttons to create point and middle point are grouped into a menu with an icon of a “point”

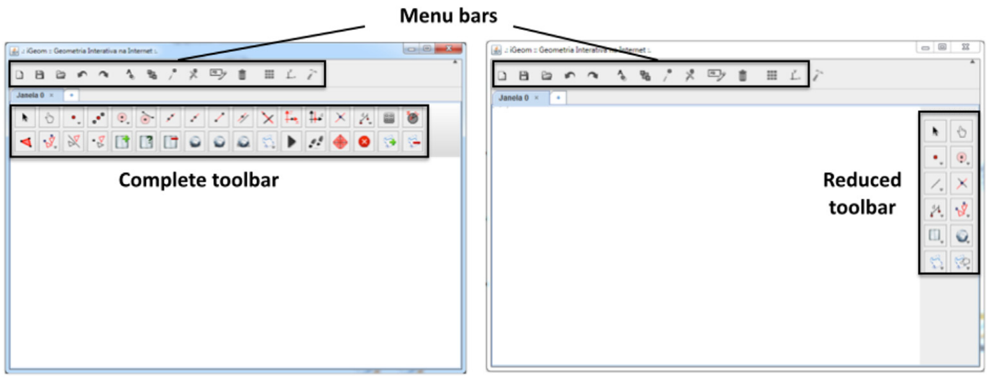

Fig. 2. iGeom's complete interface (left) and reduced interface (right).

that expands into two other buttons when users click on it. The same happens with buttons to create line, semi-lines and segments which are grouped together. The second criterion is related to the frequency of use during our previous works (Isotani & Brandão, 2008; Isotani et al., 2010). Thus, functionalities/buttons that have higher usage appears on the top of the toolbar in the reduced interface. A summary of all functionalities available in both interfaces is show on Table 1.

Finally, in the reduced interface the toolbar was moved to the right side in order to increase the drawing area. As show in Fig. 3 by moving the toolbar to the side on conventional monitors with resolution of 1024×768 it is possible to increase the drawing area in approximately 20%. Such an increase happens because most screen resolutions are bigger in width (w and w') and smaller in height (h and h').

5. Experiment setup

This section details the experiment setup we used to investigate the effects that reducing iGeom's GUI can have on usability. More specifically, we designed the experiment to examine whether iGeom's reduced interface (i.e., which displays only a basic subset of features) enhances usability in comparison to its complete interface (i.e., a version of the interface that tries to incorporate as many

Table 1
All menu tasks available in iGeom's menu bar.

Complete interface	Reduced interface	
Button function	*Button function*	*Drop-down button*
1. Marquee tool	1. Marquee tool	
2. Move point	2. Move point	
3. Create point	3. Point	Create point
		Create midpoint
4. Create midpoint		
5. Create circumference (given two points)	4. Circumference	Create circumference (given two points)
6. Create circumference (from point and segment)		Create circumference (from point and segment)
7. Create line	5. Line	Create line
8. Create ray		Create ray
9. Create segment		Create segment
10. Create parallels		Create parallels
11. Create line perpendicular		Create line perpendicular
12. Create Cartesian axis		Create Cartesian axis
13. Create line perpendicular to the axis		Create line perpendicular to the axis
14. Intersection	6. Intersection	
15. Measure	7. Measure	Measure distance
16. Calculator		Calculator
17. Measure circular arc		Measure circular arc
18. Build interior and integral/area		Build interior and integral/area
19. Isometrics	8. Isometrics	Translation move
20. Translation move		Rotation move
21. Rotation move		Reflection move
22. Reflection move		
23. Create exercise	9. Exercises	Create exercise
24. Asses		Asses
25. Undo exercise		Undo exercise
26. Communication (.GEO)	10. Communication	Communication (.GEO)
27. Communication (.html)		Communication (.html)
28. Communication (.src)		Communication (.src)
29. User-defined scripts buttons	11. User-defined scripts buttons	
30. Play scrip	12. Scripts	Play scrip
31. Recurrence		Recurrence
32. Step-by-step		Step-by-step
33. Cancel		Cancel
34. Add script		Add script
35. Remove script		Remove script

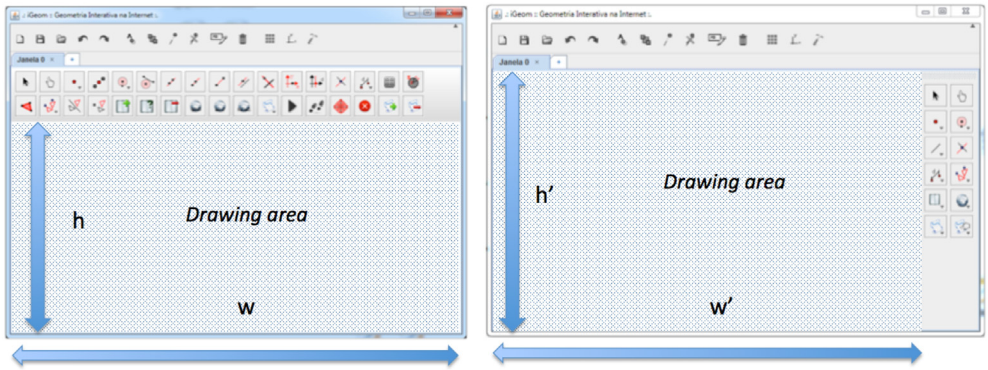

Fig. 3. Comparisons of iGeom's drawing area with resolution of 1024×768.

features as possible, exposing the users to the full functionality/complexity of iGeom). Fig. 2 shows the complete and reduced interfaces. Our research question (**RQ**) can be formalized as the following:

RQ$_1$: In terms of usability, how much can iGeom's interface be improved when reduced to its basic features?

The experimental design we used to evaluate this question is five-fold, comprising the following steps: (i) personal questionnaire, (ii) pretest, (iii) intervention, (iv) posttest, and (v) usability questionnaire.

5.1. Goal definition

We used the organization proposed by the Goal/Question/-Metric (GQM) paradigm (Wohlin et al., 1999), describing our experimental goals in five parts: object of study, purpose, perspective, quality focus, and context.

- **Object of study**: The objects of study are two versions of iGeom's GUI, namely, reduced and complete.
- **Purpose**: The purpose of this experiment is to evaluate two approaches to presenting features in IG software with respect to their benefits in terms of usability. Specifically, we investigate whether a reduced interface is a significant improvement over the complete interface, which shows advanced and extraneous graphical elements. The experiment provides insight into how much usability is enhanced by either hiding or disabling extraneous graphical elements. It is also expected that the experimental results can be used to bolster up the CLT. In this case, our reduced interface conforms to CLT concepts by minimizing cognitive load and thus enhancing working memory-related processing. On the other hand, the complete version of iGeom's GUI shows extraneous elements that may interfere with the process of acquiring knowledge and skills.
- **Perspective**: This experiment is run from the standpoint of a researcher.
- **Quality focus**: The primary effect under investigation is usability as measured by the score obtained by the subjects using either of two versions of iGeom's GUI.
- **Context**: This experiment was carried out using 69 undergraduate students as subjects. We followed a randomized trial design as follows: a group of 38 subjects participated in the experiment by using the reduced interface, while the other group, containing 31 subjects, used the complete interface. The experiment was carried out in the facilities of the University of São Paulo. All students that participated in the experiment signed up for a computer science course. More specifically, all subjects that participated in this experiment were enrolled in a Software Engineering course.

Our experiment can be summarized using the template by Wohlin et al. (1999), as follows:

Analyze iGeom's reduced and complete interfaces

for the purpose of comparing

with respect to their usability

from the point of view of the researcher

in the context of computer science undergraduate students with little or no background knowledge in IG software and geometry.

5.2. Hypothesis formulation

We formalized our research question (RQ$_1$) into hypotheses so that statistical tests could be carried out:

Null hypothesis, H$_0$: There is no difference in usability between the two interfaces (measured in terms of the score achieved by the subjects in the pretest and posttest), which can be formalized as:

H$_0$: μreduced interface $=$ μcomplete interface

Alternative hypothesis, H$_1$: There is a significant difference in usability between the two versions of the interface (measured in terms of the score achieved by the subjects in the pretest and posttest):

H$_1$: μreduced interface $\neq$ μcomplete interface

5.3. Experiment design

Aimed at verifying our conjecture, we applied a standard design with one factor and two treatments (Wohlin et al., 1999). The main factor of the underlying experiment, an independent variable, is usability. The treatments or levels of this factor are the two interfaces, namely, reduced and complete. In this experiment setup, the main dependent variable is the scores of the subjects, which is defined as the number of geometry questions that they correctly answered using iGeom's features. Both data from the pretest and posttest were classified as the following: 3 points for a completely correct answer; 2 points for answers containing minor problems; 1 point for answers whose lack of geometry knowledge on the part of the subject thwarted a more proper answer; and no points for utterly wrong answers.

Both pretest and posttest encompassed 20 geometry questions. Each of these sets included six easy questions, seven intermediate questions, six advanced questions, and a very advanced question. It is worth noting that the complexity of these questions is not directly related to geometry concepts. Rather, the more the user has to interact with iGeom's GUI, the more complex the question according to our criteria: tougher questions require more mouse clicks to be solved. We devised these questions in hopes of evaluating the user's interaction with a reduced and a complete GUI. That is, we were not interested in evaluating the user's proficiency in geometry. Table 1 shows the mean amount of mouse clicks assigned to questions according to our previous studies (Isotani & Brandão, 2008; Isotani et al., 2010; Reis et al., 2012). As shown in Table 2, easy questions require around three to eleven mouse clicks in the reduced interface and two to eight clicks in the complete interface. Moreover, as highlighted in the Table, as difficulty goes up, the amount of mouse clicks also increases.

Fig. 4 shows an easy question to build a middle point using two points *A* and *B* available in the drawing area of the reduced interface. To solve this question, users have to press the "Point" button (Figs. 4-1). After clicking on this button, users have to select the option "midpoint" (Figs. 4-2). And then, select both points *A* and *B*, in any order (Figs. 4-3 and 4-4). These actions will create the middle point *C*. Thus, the minimal amount of mouse clicks required to solve this question is four, which means that this is an easy question according to the mean amount of clicks shown in Table 1. In the complete interface, an user would take 3 clicks (one less click than in the reduced interface) since the middle point button is directly available in the toolbar.

An intermediary task is shown in Fig. 5 where a user has to create a bisector line using the given points *A* and *E*. In the Figure each number in parenthesis represents the number of clicks of an action. To complete the task, the user has to click and select the option to generate a circumference using the point A (Fig. 5-1 to 5-3). The same also applies for point E (Fig. 5-4 to 5-6). Afterwards, two intersection points have to be created at the point in which the two circumferences overlap (Fig. 5-7 to 5-10). Finally, these intersection points have to be selected and the user has to select the option to draw a line (Fig. 5-11 to 5-13). This question required thirteen mouse clicks in the complete interface. The same construction would require 18 clicks to be completed in the reduced interface, since one extra click is required for every click made in the toolbar of the complete interface.

Next, we present further information about the evaluation we carried out.

5.3.1. *Personal questionnaire*

In the first step, the subjects were required to fill out a questionnaire on their knowledge of geometry concepts, background with IG software, and some personal information. The purpose of this questionnaire was to classify the subjects according to their technical and schooling level. From the questionnaire, we found that most subjects have advanced computer skills. Furthermore, most subjects stated that they have a good grounding in geometry.

Table 2
Mean amount of mouse clicks needed to solve varying complexity questions.

Difficulty level	Reduced interface	Complete interface
Easy	3–11	2–8
Intermediary	12–19	9–14
Hard	20–35	15–30
Very hard	36–50	31–45

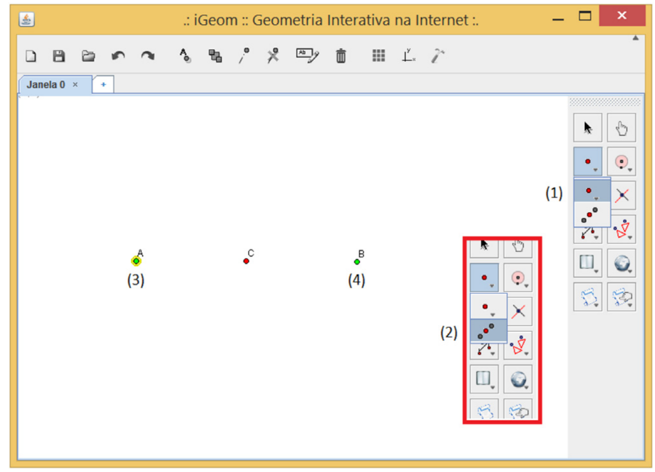

Fig. 4. Amount of mouse clicks needed to solve an easy question using the reduced interface.

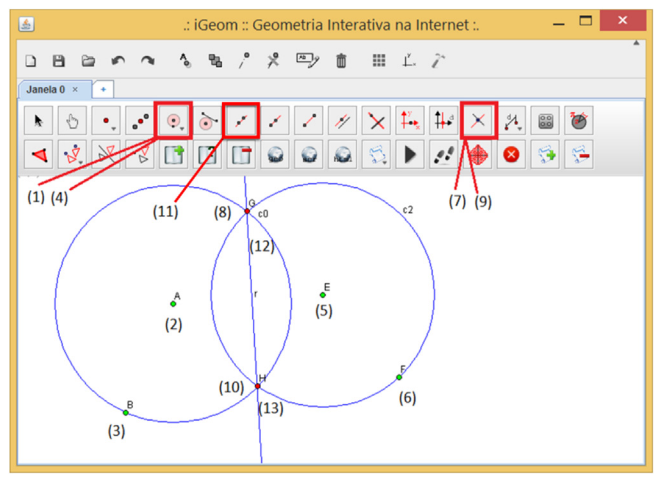

Fig. 5. Amount of mouse clicks needed to solve an intermediate question in the complete interface.

Only four students had prior experience with IG software. Only one student had prior experience with iGeom.

5.3.2. *Pretest*

During this step, we presented the software to the subjects. In order to gauge the proficiency of the subjects in using each GUI for the first time, at first there were no preliminary instructions on how to use either interface. Thus, we were able to observe the behavior of each subject and how they tried to solve each question throughout the tests. It is worth mentioning that the subjects were allowed to search the Internet to better grasp geometry concepts to complete the given tasks.

The subjects had up to 1 h to solve 20 questions in pretest. Each question was graded from 0 (totally wrong) to 3 (totally right) in order to determine each participant general score. Subjects interacting with iGeom through its reduced interface had an average score of 28.53 out of 60 points. Subjects using the complete interface achieved an average score of 25.61. Table 3 summarizes the scores obtained during the pretest. From analyzing that data shown in the Table it can be seen that the subject that achieved the highest score used iGeom's reduced interface. However, a subject within the reduced interface group also obtained the lowest score.

Table 3
Descriptive statistics summarizing the scores achieved by the subjects in the tests.

	GUI version			
	Pretest		Posttest	
Descriptive statistics	Complete	Reduced	Complete	Reduced
Max	44	47	42	41
Min	16	14	17	12
Mean	25.61	28.53	30.13	26.82
Median	24	29.5	30	27
Standard deviation	6.22	7.04	5.88	5.41

Fig. 6 shows boxplots giving an overview of the scores achieved by each group. Initial analysis of the pretest information in Table 3 and the boxplots in Fig. 6 suggests that subjects using the reduced interface performed better than the subjects using the complete interface, obtaining higher overall scores: the mean of completely correct answers for subjects using the reduced interface was 7.47 (which would result in a score of 21 points). Subjects in the complete interface group achieved an average of 6.32 completely correct answers (18 points).

5.3.3. Intervention

After the pretest, both groups of subjects took part in a series of video lessons. These video lessons were tailored toward describing each feature implemented by iGeom, giving examples and usage scenarios. Moreover, during those video lessons, the subjects were walked through how to solve the questions presented in the pretest.

5.3.4. Posttest

After going through the intervention step, the subjects took the posttest. The contents of the posttest are similar to the pretest: it also contains 20 questions with varying difficulty levels. By examining the data in Table 3 we can see that the subjects using the complete interface performed better than the subjects using the reduced interface. The complete interface group obtained a mean score of 30.13 points while the reduced interface group achieved a mean score of 26.82. Furthermore, the highest score was obtained from a subject in the complete interface group: 42 points (Table 3). However, the highest score among the subjects using the reduced interface was quite similar to the highest score: 41 points (Table 3). On average, subjects in the complete interface group got 9.52 completely correct answers, whereas the reduced interface group scored an average of 8.16 completely correct answers. Fig. 7 shows boxplots giving an overview of the scores achieved by each group during the posttest.

5.3.5. Usability questionnaire

After taking the posttest, we had the subjects answer a questionnaire with the purpose of assessing the users' opinions about iGeom. This questionnaire contains 15 questions, whose answers employ a linear scale that goes from 1 to 5: where 1 implies "Strongly disagree" and 5 "Strongly agree". Table 4 gives an overview of the three questions that achieved the most "Strongly agree" answers. As shown in Table 4, according to most users, the captions of most buttons should better reflect their functionality. In addition, the results suggest that most users are not satisfied with the icons/images used in these buttons, and that the layout of these button should change. Given the results in Table 4, future work will now focus on revamping iGeom GUI.

5.4. Analysis of data

This section presents our experimental findings based on the results described in the previous sections. We divided the analysis into two topics: discussion and hypothesis testing.

5.4.1. Discussion

This section analyzes the results presented in the previous sections. From Fig. 6, it can be seen that the subjects in the reduced interface group outperformed the complete interface group in the pretest, obtaining relatively higher scores. This suggests that for novice IG software users, a clear-cut, reduced interface is more

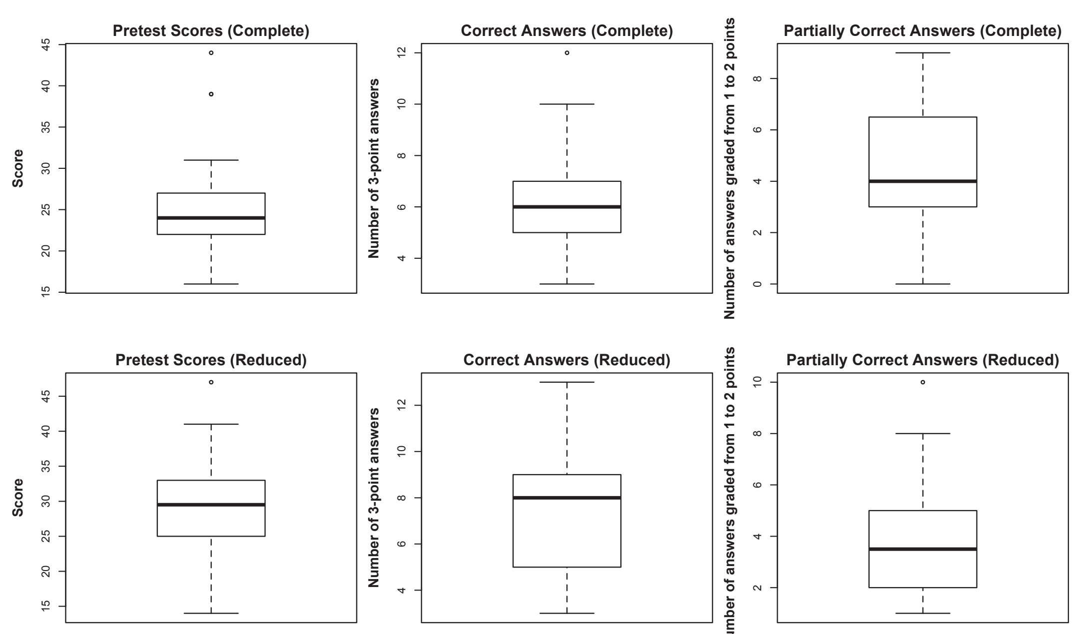

Fig. 6. Boxplots summarizing the data from the **pretest**, from left to right: the scores obtained by the subjects that used either the complete or the reduced interface, the number of questions that were graded as correct (3 points), and the number of partially correct questions, i.e., questions that were graded from 1 to 2 points.

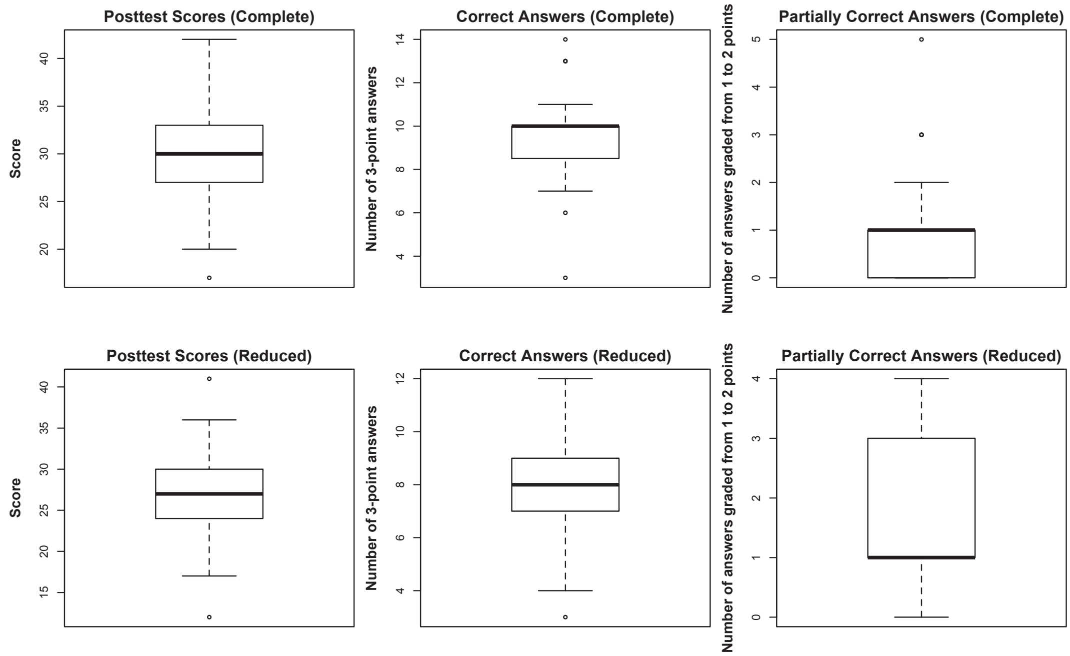

Fig. 7. Boxplots summarizing the data from the **posttest**, from left to right: the scores obtained by the subjects that used either the complete or the reduced interface, the number of questions that were graded as correct (3 points), and the number of partially correct questions, i.e., questions that were graded from 1 to 2 points.

Table 4
The three major issues according to the subjects: Captions' meaning, icons' quality and buttons' layout.

	Reduced interface user's answers	Complete interface user's answers
Button's captions should better explain buttons' functionality	75%	76%
Buttons' icons should display more meaningful images	30%	42%
Buttons' layout should be rearranged	21%	15%

effective, since when many icons with similar functions are present together; it becomes more difficult for the user to distinguish the desired function among them. However, the reduced GUI keeps only icons with more distant functions, therefore making it simpler for the user to distinguish which icon is related to a given function. This is in line with CLT, which contends that the amount of information that subjects have to process can overload their working memory, thereby hindering learning. However, after going through the intervention step (Subsection 4.5), it seems that subjects using the complete interface achieved slightly higher scores (Table 3 and Fig. 7).

The data dispersion shown in Table 3 (i.e., standard deviations) indicates that subjects in both groups performed in a consistent fashion in the posttest. Presumably, during the posttest, most of the questions left unanswered are due to the subjects' lack of geometry knowledge and not problems with the underlying interface. It is also worth mentioning that since most subjects were unable to finish neither the pretest nor the posttest, it might be useful to enlarge the time window in future experiments.

5.4.2. Hypothesis testing

Aimed at testing the hypotheses in Subsection 4.2, we performed a Mann–Whitney U test, which is a non-parametric test used to compare the medians of two independent samples (Siegel & Jr., 1988). The Mann–Whitney test has power-efficiency of about 95% of the t test on two samples from normally distributed populations of equal variance of scores which were independently drawn. But if the two tests were both used with data from non-normal populations or with data from populations differing in variance, the Mann–Whitney test might very well reject $\mathbf{H_0}$ at a more stringent significance level than would a t test (Siegel, 1957, p. 18).

When comparing the pretest scores of the reduced and complete groups, we obtained the distributions in the two groups differed significantly (Mann–Whitney U test $U = 389$, $z = -2{,}146$, $n_1 = 31$ $n_2 = 38$, $P = 0.016$ two-tailed). This result indicates that we can reject H_0, that is, a reduced interface is more helpful than an interface containing extraneous and advanced features. In a future work, we will try to compare our data with results from an experiment that takes subjects with previous geometry knowledge into account.

The results of the posttest was $U = 382$, $z = -2{,}497$, $n_1 = 31$ $n_2 = 38$, $P = 0.013$ two-tailed. Since the p-value is smaller than 0.05 we can refute H_0, backing up the alternative that states there is a significant difference in usability between the two versions of the interface (measured in terms of the score achieved by the subjects in the pretest and posttest). Therefore, we can conclude that after receiving proper training, a complete interface is better than a reduced one in the context of IG software.

6. Future work

We intend to carry out follow-up studies to gather more information about the students' proficiency in Mathematics and its correlation on students' ability to use the GUI of GI software. We plan to introduce the following steps in follow-up experiments:

students/subjects will (i) fill a questionnaire with personal questions, (ii) take a test tailored to gauge their computational skills, (iii) take a geometry test, and (iv) have their working memory evaluated through some standard test.

In order to evaluate the feasibility of the aforementioned steps and in an attempt to enhance the experiment design prior to carrying out a full-scale investigation, a pilot study was conducted. The subjects of this pilot study were seven graduate computer science students (Borges, Reis, Moro, Durelli, & Isotani, 2013). The experiment setup evaluated a single factor with only one treatment for each variable. The main independent variables were working memory (WM), computational knowledge (CK), and geometry knowledge (GK). The result of each of these variables was compared, in turn, with the main independent variable: the score achieved by the subject when trying to solve a couple of geometry questions using iGeom. The test to evaluate the computational proficiency of the subjects encompassed questions on basic knowledge of computer usage. The geometry test included multiple-choice questions whose levels ranged from elementary to advanced. To evaluate the working memory of the subjects, we applied n-back tests (Dehn, 2008).

The questionnaires answers indicated that the subjects have advanced computational knowledge; however, our experiment setup was unable to determine whether this knowledge benefits novice users when dealing with an IG as iGeom. However, these preliminary results indicate that geometry knowledge may be a key factor that makes using a tool such as iGeom more approachable. Most participants achieved high scores on the geometry test; however, none of them achieved the maximum score.

Our results evince that another key factor that plays an important role in learning a new tool is working memory. According to our results, it seems that 11% of the final score using iGeom was influenced by the subjects' working memory (Borges et al., 2013). It is important to emphasize that we did not remove outliers because this would have an adverse effect on the amount of data available. In addition, it is hard to draw substantial conclusions from a pilot experiment involving only seven participants. However, the main purpose of carrying out this pilot (i.e., improving the experiment design before conducting a full-scale experiment) study was achieved.

We also intend to carry out experiments involving more subjects to consider other relevant aspects of a GUI such as icon metaphors and affordance that are also important variable to understand the impact of interface on students' learning and interaction with GI software.

7. Concluding remarks

A GUI is a key element that dictates the interplay between users and software systems. Since interfaces define the boundaries through which users interact with software systems, they play quite an important role in the context of educational software. Thus, it is paramount to evaluate which characteristics GUIs for educational software should have. To this end, we investigated whether a reduced interface made few cognitive demands on users in comparison to a complete interface. In doing so, we designed two interfaces for the IG software and carried out an experiment involving 69 subjects. We found that, at first, users are slightly more productive when using a reduced interface. However, after being familiarized with the underlying educational software, users of the complete interface outperformed the ones using a reduced interface. Therefore, we conclude that a clear–cut interface is more amenable to novice users, and after overcoming the initial learning curve, a complete interface can make users more productive.

Acknowledgments

The authors would like to thank the financial support provided by CNPq, CAPES and FAPESP. We are also thankful to Aline A. Santos, who helped us to run the experiment.

References

Baker, K., Greenberg, S., & Gutwin, C. (2001). Heuristic evaluation of groupware based on the mechanics of collaboration. In *Proceedings of the 8th IFIP International Conference on Engineering for human-computer interaction* (pp. 123–140).

Borges, S. S., Reis, H. M., Moro, L. F., Durelli, V. H. S., & Isotani, S. (2013). Agrupamento Balanceado de Sujeitos a fim de Testar a Interface Gráfica de um Software de Geometria Interativa. In *RENOTE – Revista Novas Tecnologias na Educação* (Vol. 11, pp. 1–10).

Chandler, P., & Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognition and Instruction, 8*(4), 293–332.

Coyne, J. T., Baldwin, C., Cole, A., Sibley, C., & Roberts, D. M. (2009). applying real time physiological measures of cognitive load to improve training. In *Proceedings of the 5th International Conference on Foundations of Augmented Cognition* (pp. 469–478). Neuroergonomics and Operational Neuroscience, Springer-Verlag.

Dehn, J. M. (2008). *Working memory and academic learning: Assessment and intervention.* John Wiley & Sons.

Erbas, K., & Yenmez, A. A. (2011). The effect of inquiry based explorations in a dynamic geometry environment on sixth grade students' achievements in polygons. *Computer & Education, 57*(4), 2462–2475.

Erez, M. M., & Yerushalmy, M. (2006). If you can turn a rectangle into a square, you can turn a square into a rectangle... young students experience the dragging tool. *International Journal of Computers for Mathematical Learning, 11*(3), 271–299.

Findlater, L., & McGrenere, J. (2007). Evaluating reduced functionality interfaces according to feature findability and awareness. In , *Lecture notes in Computer Science: Vol. 4662. Proceedings of the IFIP International Conference on Human-Computer Interaction (INTERACT)* (pp. 592–605).

Gog, T. van, & Paas, F. (2012). Cognitive load measurement. In N. M. Seel (Ed.), *Encyclopedia of the sciences of learning* (pp. 599–601). US: Springer.

Hollebrands, K. F. (2003). High school students' understandings of geometric transformations in the context of a technological environment. *The Journal of Mathematical Behavior, 22*(1), 55–72.

Isotani, S., & Brandão, L. O. (2008). An algorithm for automatic checking of exercises in a dynamic geometry system: iGeom. *Computer & Education, 51*(3), 1283–1303.

Isotani, S., Mizoguchi, R., Inaba, A., & Ikeda, M. (2010). The foundations of a theory-aware authoring tool for CSCL design,. *Computers & Education, 54*(4), 809–834.

Kortenkamp, U., & Dohrmann, C. (2010). User interface design for dynamic geometry software. *Acta Didactica Napocensia, CADGME, 3*(2), 59–66.

Laborde, C. (2007). The role and uses of technologies in mathematics classrooms: between challenge and modus vivendi. *Canadian Journal of Science, Mathematics and Technology Education, 7*(1), 68–92.

Mackrell, K. (2011). Design decisions in interactive geometry software. *ZDM, 43*, 373–387.

Nielsen, J. (1993). *Usability engineering*. Morgan Kaufmann Publishers.

Oliveiras, L. L., & Santos, E. T. (2003). A study to evaluate the usability of iGeom graphical user interface. In *V International Conference on Graphics Engineering for Arts and design (GRAPHICA)* (pp. 1–9).

Paas, F., Renkl, A., & Sweller, J. (2003). Cognitive load theory and instructional design: recent developments. *Educational Psychologist, 38*(1), 1–4.

Plass, J. L., Chun, D. M., Mayer, R. E., & Leutner, D. (2003). Cognitive load in reading a foreign language text with multimedia aids and the influence of verbal and spatial abilities. *Computers in Human Behavior, 19*(2), 221–243.

Reis, H. M., Borges, S. S., Durelli, V. H. S., Moro, L. F., Brandão, A. A. F., Jaques, P., et al. (2012). Towards reducing cognitive load and enhancing usability through a reduced graphical user interface for a dynamic geometry system: an experimental study. IEEE International Symposium on Multimedia, 2012, Irvine. In *Proceedings of the IEEE International Symposium on Multimedia* (pp. 445–450).

Reis, H. M., Borges, S. S., Griffiths, P. A., Moro, L. F., & Isotani, S. (2013). Interação em Interfaces de Softwares de Geometria Interativa: um Mapeamento Sistemático. In *Simpósio Brasileiro de Informática na Educação 2013 in Anais do Congresso Brasileiro de Informática na Educação, 2013* (pp. 1–9).

Richter-Gebert, J., Kortenkamp, U., & Ulrich, H. (2012). *Working with the interactive geometry software*. Springer.

Roanes-Lozano, E. (2003). A bridge between dynamic geometry and computer algebra. *Mathematical and Computer Modeling, 37*(9–10), 1005–1028.

Schimpf, F., & Spannagel, C. (2011). Reducing the graphical user interface of a dynamic geometry system. *ZDM, 43*, 389–397.

Sedig, K., & Liang, H.-N. (2006). Interactivity of visual mathematical representations: factors affecting learning and cognitive processes. *Journal of Interactive Learning Research, 17*(2), 179–212.

Shimomura, Y., Hvannberg, E., & Hafsteinsson, H. (2013). Haptic cues as an utility to perceive and recognize geometry. *Universal Access in the Information Society, 12*(2), 125–142.
Siegel, S. (1957). Nonparametric statistics. *The American Statistician, 11*(3), 13–19.
Siegel, S., & J. C, N., Jr. (1988). *Nonparametric statistics for the Behavioral sciences* (2 edition). Boston, Mass: McGraw-Hill Humanities/Social Sciences/Languages.
Sweller, J., Ayres, P., & Kalyuga, S. (2011). *Cognitive load theory*. Springer.
Tidwell, J. (2011). *Designing interfaces* (2nd ed.). O'Reilly.
Wohlin, C., Runeson, P., Höst, M., Ohlsoon, M. C., Regnell, B., & Wesslén, A. (1999). *Experimentation in software engineering: An introduction*. Kluwer Academic Publishers.