Using Interactive Digital Learning Systems to improve STEM education: a case study using the SOWISO platform

Thesis report submitted in partial fulfillment of the requirements

for the degree of

Msc Information Science

by

Naseya Weinchard

(Roll: 2622753)

Under the supervision of

Dr. Sieuwert van Otterloo
Dr. Ana Lucia Vargas Sandoval

Computer Science
University
July 24, 2022

Abstract

ICT implementation in STEM education is often not efficient enough to improve the educational experience of students. Teachers and course creators do not have the time, or the required skill set to improve digital learning tools. Such an online environment has the potential to provide an unlimited amount of practice opportunities for students. Previous research has emphasised the downside of ICT not being correctly used in STEM education. This thesis focuses on the improvement of applications that can provide interactive learning exercises. By executing a case study, several learning formats are compared to each other, while keeping track of the author's contribution. During an internship at the company SOWISO, which provides an online platform for STEM education, I experienced the complex role of an author, with the purpose to improve the effectiveness of interactive practice tools. The exercise sets created during this internship were used for an experiment: four test groups were supposed to complete an exercise set containing first order logic content, where every group received a different format of the same exercises: progressive, anti-progressive, arbitrary and progressive with support. The analysis of the conducted results is mostly based on participants' total score, their survey answers and the contribution of an author. The outcome showed that the progressive format, the most classic educational system, scored less compared to the other three test groups.

Keywords— STEM education, ICT, digital learning systems, internship, education, author, progressive, anti-progressive, first order logic

Chapter 1

Introduction

This research proposal contains the description of research on digitising an undergraduate level introduction course on first order logic, named Skills For AI. Firstly, the problem and the motivation to investigate this problem shall be introduced. The contribution of this research will subsequently be explained, from both a scientific and practical perspective. The introduction is then followed by a brief overview of the literature that is the foundation of this research proposal; the literature is intended to be applied to the research methods and theories, in order to complete valid scientific research.

Besides the theoretical part, an experiment was executed. The methods that have been used to complete this experiment are discussed and explained. This includes participation in an internship at the company SOWISO, a company that provides a digital platform for interactive mathematical courses. Finally, the analysis of the results is used to conclude on the experiment and the corresponding theory.

1.1 Motivation

1.1.1 Skills for AI

Digital technologies nowadays are the main pillar for students to complete their education. Their development is affected by the implementation of digital tools, especially at the university level. However, it has appeared that digital development for teaching purposes is often not included, due to the lack of professional development for teachers and the minimal institutional involvement. [1] The scarcity of skill sets seems to be an issue that must be handled by educating these teachers. Further, it could be a possibility to adjust the digital environment in such a way that is more accessible for the teachers that do not yet operate on a professional digital level. Such an enhanced environment could result in an automated digital learning experience for students, with minimal effort from a teacher.

As a computer science student with a beta-tutoring background, I have always been keen to improve the educational system. After gaining quite some knowledge of computer and information sciences, the next logical step seemed to be the application of both disciplines. Improving a whole educational system is of course an impossible goal for the first step; therefore I intend to focus on implementing a particular subject in the course Skills for AI. This will result in a more detailed case study, which can be completed within six months.

Skills for AI is a course where students refresh and supplement their knowledge of mathematical topics, as their background in mathematical skills is often limited. Such a skill set is relevant due to the understanding of Artificial Intelligence (AI). This includes linear algebra, logic and probability. Therefore, the students must be able to understand and apply logical and mathematical languages, which they then have to use to express AI applications. The logic content of the course consists of both propositional and first order logic, containing the following subjects: Boolean operators, semantic equivalence, functional completeness, predicates and their quantifiers, logic models, and their corresponding necessary definitions and concepts. During this research, I chose to focus on the functionality of digitised logic models, which is a difficult element of logic theory. [2]

Linear Algebra contains topics such as matrices, linear equations, (in)dependence, eigen vectors, and basic math skills. Once again, these skills will be necessary in the application of modelling computer science and AI problems. Examples of certain problems are database recognition and neural networks. Probability theory is also discussed since this is fundamental to every academic major to complete their research. [3, 4]

1.1.2 SOWISO

Mathematical courses seem to be part of several beta degrees as a mandatory part for students to complete their degrees. Students often struggle with these mandatory parts and SOWISO succeeds in preventing this failure clearly and functionally. This is a digital learning platform which allows students to practice math exercises that are repetitive and randomised. SOWISO has several question options, such as multiple-choice, open-ended and essay questions, among others. While the students are considering their answers, they have the opportunity to request a hint, which will help them with formulating the correct answer. Whether students are able to request a hint, which is sometimes based on the information they have already entered, depends on SOWISO's implementation of the relevant exercise.

The SOWISO platform is meant for students to do their homework and prepare for exams. They should be able to receive an unlimited amount of variations on one exercise, in order to increase their practice possibilities. Besides that, teachers are also able to benefit from the functionalities as well. It is possible for teachers to use the platform as a complete course. The teaching mode enables a predetermined curriculum, provided by SOWISO. This can be compared to a digital and interactive textbook, which can be customised in every possible sense, such as adding and deleting courses, emphasising sections, and creating intermediate texts.

Languages

Furthermore, different program languages are accepted and at the same time required to create exercises. Latex is the language used for mathematical elements in a text document. If Latex is not used, then the mathematical equations do not appear in the desired way. Maxima is another application that requires its language. Deriving answers from mathematical equations is often handled by the functionalities of Maxima, which again requires another implementation. [5, 6]

1.1.3 Background

Previous research was done by Daryl Zandvliet on effective learning analytics for higher education. A majority of universities include digital online learning tools. The purpose of these platforms is often to support communication and practice opportunities. [7] Another benefit of using such an online system is the data that can be collected from students' results. Learning analytics is the term that defines the use of such conducted data. Daryl Zandvliet focused on making this process more effectively, in order to enhance the learning experience of students.

Online platforms often have implemented statistics, presented on a learning analytics dashboard. This gives teachers the possibility to understand a student's behaviour. Such a dashboard provides six different types of data sources, including learning artefacts by the students, direct information from the learners, institutional database records, system logs, and physical user activity. Besides these data sources, there are individual and class indicators, which provide detailed information on their actions, results, context and their learning curve. In total, these categories contain about 200 indicators, as defined by Schwendimann et al. [8]

During the thesis of Daryl Zandvliet, the teacher's experience was examined, as well as their beliefs about an online platform with learning analytics. About 40% of teachers stated that they did not use the dashboard functionalities after the course was finished. They were not familiar with the statistics, or they did not have access to this part of the application. Other reasons for teachers not to use the analytics, were the fact that other staff members fulfilled the task, they tried to protect student data or they did not find the data relevant enough. Another question asked during that survey, extracted alternative types of analytics used to improve learning and teaching activities. The majority of the teachers tend to use surveys as a primary tool. Examples of these alternatives are quizzes and polls, pass rates, and evaluations after the course.

Interesting about this research is the outcome that teachers tend not to use the built-in analysis. This analysis should be a benefit to digital learning environments. This suggests that teachers in general draw back from these relatively new tools, as long as they are not as efficient as their current method. In order for a new tool to become relevant and efficient, a teacher should probably be more integrated in the latest technical developments from the online educational applications.

1.1.4 Terminology: logic and learning systems

Logic is the study of correct reasoning, often used to understand what is correct reasoning in mathematics. It is often defined in a more narrow sense as the science of deductively valid inferences or of logical truths. In this sense, it constitutes a formal science investigating how conclusions follow from premises in a topic-neutral way or which propositions are true only by virtue of the logical vocabulary they contain. First order logic, also known as predicate logic is an extension of the simple propositional logic, which was explained in theory 1. This extension makes use of quantifiers and relations over predicates.

First order logic, also known as predicate logic is an extension of the simple propositional logic, which was explained above. This extension makes use of quantifiers and relations over predicates. Models are formal representations of the interpretation of first order logic formulas. In simple words, an interpretation (represented by a model) of a first-order formula specifies what each predicate means, and the entities that can instantiate the variables. What is the purpose of drawing a model for a corresponding formula? A formula could be true or false, meaning that it is either possible or impossible to find a model for it. Thus, finding a model for a given formula guarantees that there is an interpretation for which the formula is true.

Another important aspect of this research is a learning environment, which could be a school, university, community, museum, and many other circumstances. These domains share the property that they provide opportunities for learning and socialization. Now that technology is making its way into nowadays society, digital tools have entered our pool of learning environments. Computer games, social media applications, Wikipedia, and other open sources encourage individuals to learn about new content, which causes them to adopt new behaviour patterns. Digital learning systems in general are the source of design and development. [9, 10]

Several digital learning methodologies can be applied at university courses. According to Maria Sousa et al., examples of these different approaches are: project based-learning, problem-based learning, digital stories, online learning environments, digital moments, technology-integrated teaching methods, digital storytelling, educational games, and authentic learning. [11] SOWISO belongs to at least one of these examples, since this is an online learning environment. Technology integrated teaching methods also occur in the system. The platform is able to provide integrated hints, adaptive hints, control exercise order and much more. Hence, these features replace the proceedings of a teacher who usually provides hints on the stand, and tells students which exercises to do in what order.

1.2 Problem definition

Logic theory consists of complex equations and expressions and thus it is quite a challenge to make such a course interactive and digital, simultaneously. Besides this complexity, it is also a huge challenge to find out what format applies best to students their learning curve during class. There are different applications of digitised learning systems. Examples of such interactive applications are the implementations of intermediate tips and tricks or the option to control the workflow of a student where they must finish an exercise before starting another exercise. However, do we know what applications are the most beneficial for all parties, consisting of the student, teacher and content creator?

Hence, this research focuses on the possibilities of digitising a beta course, which includes logic theory from the course Skills for AI. The challenge here is to find the most efficient way of applying a digital platform to this course, considering the implementation and format of a digital logic exercise set, the results from the students and the analysis on students' results, considering their background.

1.3 Research question(s)

It is important to consider the different applications of digital interactive learning algorithms, to define the best practising formats. In order to provide and apply a product that could be used during an undergraduate graduate-level STEM education course, this research is first aiming which features of digital education platforms are important for creating undergraduate level first order logic material. Besides, the role of an author is one of the main factors in this research. Therefore, it is inevitable to consider the challenges that might occur in the process of an author. These are the (sub-)questions that shall be answered after completing this research:

(1) What would be the best format for students be to practice first order logic?

(2) What are the main challenges for authors to create undergraduate-level first order logic courses?

(3) Which features of digital education platforms are important for creating undergraduate level first order logic material?

1.4 Scientific and practical contribution

This research would provide an efficient method on applying a digital platform during the lectures of an academic level mathematics course. If this method is completed and useful, then it has the potential to be part of an undergraduate STEM course. This method will be tested and evaluated with the use of comparing different applications of interactive learning and with the help of test groups during a qualitative experiment. After evaluating the practical part of this thesis, it will possible to provide scientific consult for future authors concerning time consumption, and the application of digital learning systems. The most interesting consult comes from the methodology of digital learning; whether this could and should be adaptive or not, and what logic formulas have the best format to be digitised in the platform.

The practical contribution here would be the creation of an interactive digital platform for a STEM course inside the SOWISO platform. This environment gives the students the opportunity to practice in an environment that provides randomized exercises, with unlimited attempts to prepare for the final exam. The teacher can analyse the results of practice exercises and use those to improve the learning curve of the student. The most practical part consists of my internship at the company SOWISO: they are the providers of the platform that allows me to implement any interactive exercise set. [12] The company owns the rights over the designed exercise sets and is able to adjust the created content. The written codes are still available after the internship for me as an ex-author. Access to the platform, on the other hand, will be denied after six months. In order to adjust the content in the SOWISO platform, one must be an employee of the company.

Chapter 2

Related literature

2.1 Teaching methods and analysis

Several teaching methods and course structures appeared during the literature research; these were all applied at a university level and thus relevant to this research. The first interesting aspect that was introduced, is the flipped classroom method, a model that inverts the classic paradigm, where the initial content of the course is studied outside the classroom. While this content is handled further during lectures in the context of discussing problem-based exercises in an active way, this model could be relevant, considering the timing in providing interactive digital exercises to students. Such an interactive environment would cause the studying aspect outside the class to be more efficient. Besides the improved efficiency, such an interactive exercise environment tends to be easier to grasp than the assignment to read a chapter. Even more benefits come from the rewarding feature, which mostly motivates students. A student then gets to see when he or she correctly completed an exercise, as those are automatically graded. [13, 14, 15]

Another interesting approach supported by several papers was the formation of groups in either a flipped or non-flipped classroom. Students receive exercises without answers, with the assignment to solve the problems together with fellow students. These answers shall then be provided after a certain period of time. [16] Course structures are often quite classic, with approximately eight lectures and a final exam. However, there seems to be a very interesting alternative, namely gamification. This tends to especially motivate students to study for smaller rewards.[17]

Finally, interactive learning is not meant to replace a teacher; the predictable modelling should help the teacher focus more on the difficult themes. These statistics will give teachers an overview of the performances of students and this might hellp them in understanding what worked or did not work during the course. An interactive system could also be adaptive, where a student's exercises get adjusted automatically, based on their performances. [18] This application asks for a close analysis on the content of the course, usually completed by authors. The question that poses a challenge for teachers is to what extent the extracted information from the results will help them improve the application of an interactive system. DuoLingo, for example, gives a very nice review on their analysis that extract the aberrant test takers. [19]

2.2 Adaptive learning systems

One of the opportunities of digital education discussed during this research is an adaptive learning system (ALS); systems were the content is adjusted based on the input of the student. Anticipation, in this sense, means that the student may need some extra practice on a certain subject, based on the number of mistakes made during the corresponding set of exercises. The system then has the ability to provide the student with some personalised exercises in order to achieve sufficiency.

The performance of students can be improved due to the application of personalised environments; in this case, this will be an ALS. Such an improvement will show up in a learning curve, which can be used to determine the effect of the ALS. Hence, the analysis of students' results weighs at least as much as the implementation of the ALS itself. Analysis can result in improvement and a better understanding of ALS's effects. [20]

2.3 Progressive learning systems

Progressive learning and anti-progressive systems are not automatically interactive, as they do not anticipate to the students' results. A system is only adaptive if the content reacts to the performance of a student. In a progressive system, a student should pass an exercise, to be provided with an exercise set consisting of a higher difficulty. Until here, the approach is similar to an adaptive system. However, if a student fails an exercise, he or she will be stuck at that same level. In an adaptive learning environment on the other hand, a student would then be set back to a lower difficulty. According to a research by Mutasov, the progressive approach, which is classic, has the potential to almost always succeed. [21] Anti-progressive learning would then be the exact opposite, where students start with an exercise that is always harder than the next one. This method seems to be the exact opposite of the classic approach, therefore it is interesting to consider during this research.

Chapter 3

Research strategies and research methods

3.1 Methods (Internship)

SOWISO [1] is a digital education company founded in The Netherlands, that has developed a learning platform for courses in STEM education. The courses are used at high schools and universities in countries all around the world. The SOWISO organisation consists of several departments, namely: authors, development, commercial, and much more. For this research project, I did an internship at SOWISO as a(n) (junior) author. In this role, I experienced what it is like to be an author of course material.

[1] SOWISO: the company

3.1.1 Five method phases

The internship at the company SOWISO has made it possible to explore the digital authoring world, covering different implementations of mathematical exercises. My method, executed at SOWISO, can be explained and defined in five phases: integration, design, implementation, experiment, and analysis. There are a lot of different components used during such an implementation. These components consist of external mathematical applications, which require a certain skill set before actually using them for any implementation. Hence, the first phase of this internship must consist of integration. After the integration I developed the desired skill set in order to estimate the design possibilities regarding the implementation of logic models.

Phase 1: Integration: platform, languages, GeoGebra & JavaScript

Platform

The SOWISO platform consists of a complex structure and many functional features. First, I should focus on the hierarchy, which is as follows: exercise, the exercise set, the package of that exercise set, the subchapter, chapter, and course of that exercise set, from small to large respectively. Second, the focus should be on the features of the platform, such as copying, saving and deleting elements. Finally, I should deal with authority, since I am not allowed to adjust every provided exercise set. The goal here is to understand the working of the platform completely.

GeoGebra

GeoGebra is the most interesting part of the integration for the upcoming experiment; this application allows diagrams and models to be visual and interactive. Until my internship, it appeared that SOWISO had only used interactive diagrams, and not employee-implemented models. During this integration process, it became clear that it was my job to innovate a functional way of creating interactive models. The background of these interactive elements in the mathematical application can be created in JavaScript. This language is accepted by GeoGebra and allows the author to include more features. [22, 23]

Phase 2: Designing models

I created a generic setup for one type of logic exercise, that could be used to create exercises at different levels. The difficulty of the exercises shall be determined based on the content of the course Skills for AI, containing first order logic sections. Example exams were provided by the course creator, which indicates the desired level. I designed this in multiple iterations that I would show to other authors at SOWISO. Besides, I would compare the format of my exercises to already existing exercises. I used those findings and their feedback to improve the model in consecutive iterations. I also documented any platform limitations, challenges or time-consuming tasks that I experienced as an author.

Phase 3: Implementation

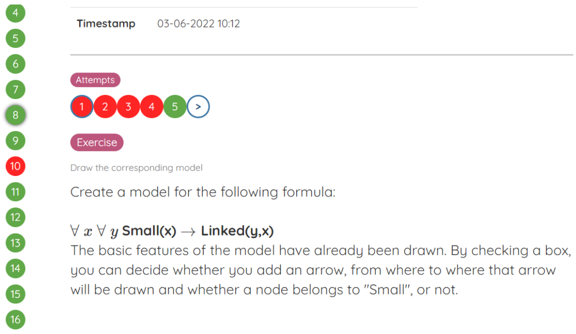
The implementation of the designed formulas and their corresponding model would consist of a few steps: Java, JavaScript, GeoGebra and SOWISO. Java will be used to derive the models that belong to a certain formula, while JavaScript is the language in which the models are defined. Defining in this sense means that the connections between the nodes of a model are specified in the code. After processing the models, the code should be included inside GeoGebra. Important here is to stick to the original naming of the GeoGebra library; this is required when the web application must read and display every element. Another important part of the implementation method, is the intermediate saving and error checking. This will prevent the author (me) to participate in any unnecessary time-consuming code backtracking. The final step should be the SOWISO implementation, considering its hierarchy and different settings for exercise sets.

Phase 4: Experiment

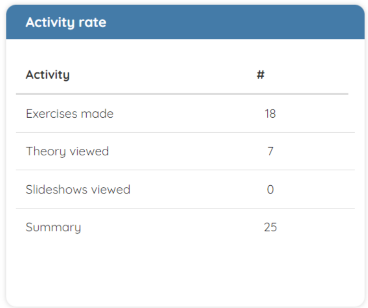
This experiment concerns a group of people; namely 40. These participants will receive a small set of exercises, that is either progressive, anti-progressive, arbitrary, or progressive with hints. Each package will cover the same subject, along with a different implementation of interactive learning: the exercises are identical with a variation from easy to very difficult. The four different classes are Alfa, Beta, Gamma, and Delta. Participants are assigned to a group based on their dates of birth; the first group includes people born in January, February and March, while the second group includes people born in April, May and June. The same system was used for groups Gamma and Delta. Participants will be recruited via my supervisor(s), inside the company, among fellow students, and my family. They have all received the same standard message containing the necessary information, without naming the exact subject of the experiment.

Every participant must read the instructions, which tell them more about the functionalities of the platform. After this section, they get access to the theory page, which provides them with the necessary concepts of propositional and first order logic. It is not possible for students to start the exercise without going through the instruction and theory page. While answering any question in the exercise set, they have unlimited access to the theory page. However, the order in which they can complete the exercises is limited: they must complete an exercise before completing the next one. This is a feature of interactive learning. On the other hand, the attempts on one exercise are unrestricted, due to the settings that I will use. If a participant does not know the answer, they must click on the "check solution"-button first. In this way, they are obligated to view the solutions, which increases the chance of them learning from the exercise. Thus, this is yet another interactive feature. Group Delta has access to hints which can help them formulate the correct answer. When accessing hints no points are deducted from the score. Participants will need approximately 30 minutes to complete the set of 10 exercises and six survey questions. These survey questions collect information about participants' knowledge of logic, their age, their response to the experiment, and their educational background.

The other aspect of the experiment was my participation during the implementation and integration. I took the role of a course creator, and kept track of all the steps that they would have to go through in order to design and implement exercises. As this is a crucial part, the results of the creator's participation is included in the analysis of the results. The results of this experiment are easy to derive as the SOWISO platform provides access to every relevant element per student, such as the number of exercise attempts, the time per attempt, and the (partial) scores. The goal is to find correlations between different group results, total scores, and the participant's background. Below you can see how the participants results are displayed. This enables me to compare the efficiency per group, mostly displayed in scatter plots. [24]



(a) Displayed results by SOWISO



(b) Summary activity

Figure 3.1: Result derivation and analysis

Phase 5: Analysis

After the results and observations are conducted, it is needed to carry out several analyses. The first step of this examination, would be the overview of the general numbers. Examples of relevant general numbers are: the total amount of participants, the average overall score, or the amount of responses to the survey questions, where a very large amount is relevant enough to highlight and examine even further. Second, I will focus on correlation 1: comparing the score on easy exercises. This set is identical to the four test groups, and is expected to at least be completed by every participant. The results will be presented in graphs, in order to make the differences visual. The second correlation, will be the analysis of the largest amount of answers from the survey question on prior knowledge (see appendix B.2). The third correlation will be based on the author's effort, concerning the time per implementation. Furthermore, it is undeniable to include the experience from the participants, also conducted from the survey questions. Finally, statistics should be applied in order to understand whether the results of tests are significant. This will be done by analysing the *p-value*. [25]

Chapter 4

Results

4.1 Phase 1: Integration

The integration at SOWISO was quite intense, due to the fact that the internship only lasted for six months. During this first period of time, it was important for me as a starter to already master the basic skills of programming. While integrating into the system, there is no room for extra studying on a certain language or a system. The most important languages were JavaScript and LateX. Another important aspect learned from being an author, is the fact that authors often have to think as a teacher as well as a student. The consideration constantly consists of for example the format that exercises should have. This is relevant due to the learning ability of the student, but at the same time the amount of effort it will take a teacher to grade the assignments. Not every exercise format is automatically graded, as essay questions.

Moreover, being a starting (junior) author at SOWISO can be hard, due to the different tools and features used in the process of creating exercises. The first step consisted of understanding the hierarchy of the platform, as explained in section 3.1.1. The smallest element is always an exercise, which much always be part of an exercise set, whereas an exercise set is always part of a package, where the package then is again part of a subchapter, etc. SOWISO maintains this hierarchy due to security reasons, considering that it is more difficult to remove and delete content from a course.

Besides this hierarchy, SOWISO uses the feature where you can copy any possible element; an exercise, but also a course. This feature allows you to copy the reference, so that any copy is always linked to the original version. Another benefit is hard copying, where the copied version of an exercise is completely independent of the original version.

Finally, the most complex part of the platform, was understanding the basics of implementing an exercise. The platform allows you to choose between standard formats of exercise types, such as multiple choice, essay questions, drag and drop, and many more. Each one requires a different way of implementation and interpretation. The implementation here entails the creation of possible answers and the grading of the correct answer(s). Understanding these features is vital for the development of an experiment design, which consists of exercises created in their platform. Such an integration makes one aware of all the possibilities, resulting in a consideration of every exercise type.

(a) Empty model (b) Model with nodes

Figure 4.1: Result derivation and analysis

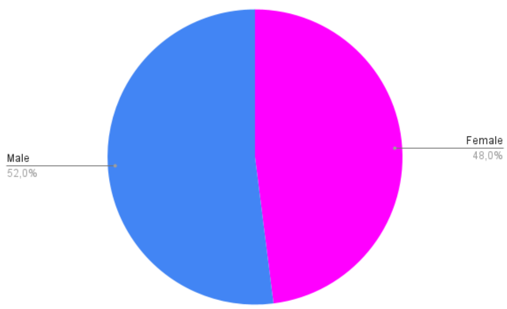
4.2 Phase 2: Designing models

Designing a model, in general, was possible at this point, as I had gained knowledge on the platform and the different possibilities combined with external applications. Understanding the possibilities and features immediately made it clear that the models could, at most, include two nodes in order to reduce the amount of possible answer combinations. The answers are designed pre-fixed, and displayed by standard check boxes with possible combinations. This seemed to be the most efficient way, considering that the answers should eventually be checked SOWISO's. An example of such an empty model is shown in figure 3.1a.

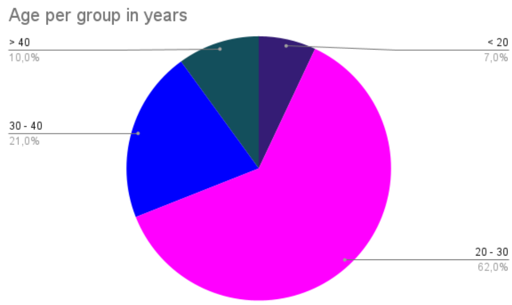
Now that the format of such a model has been created, the next step was the creation of exercise sets. This requires comprehending the difficulty and the exact subjects that should be emphasised. An exercise existed of a given formula, where a corresponding model has to be derived by the participant. Some examples of formulas have been provided by my supervisor(s). Variations in these models made it possible for me to create an exercise set. The properties of the exercise set were as follows: a variation of easy and difficult exercises, and interactive. Moreover, a reference to theory had to be included in the exercise set(s), considering that people without any knowledge of first order logic theory might participate in the experiment. It included a section discussing propositional logic, a relatively easy starter's theory.

4.3 Phase 3: Implementation

The implementation of the designed formulas and their corresponding model would consist of a few steps: Java, JavaScript, GeoGebra and SOWISO. Java will be used to derive the models that belong to a certain formula, while JavaScript is the language in which the models are defined. Defining in this sense means that the connections between the nodes of a model are specified in the code. After processing the models, the code should be included inside GeoGebra. Important here is to stick to the original naming of the GeoGebra library; this is required when the web application must read and display every element. Another important part of the implementation method, is the intermediate saving and error checking. This will prevent the author (me) to participate in any unnecessary time-consuming code backtracking. The final step should be the SOWISO implementation, considering its hierarchy and different settings for exercise sets.



(a) Male versus female participants



(b) Groups based on age

Figure 4.2: General numbers

4.4 Phase 4 and 5: Experiment and analysis

4.4.1 General numbers

In making an analysis of the results, general numbers are an interesting starting point. Some of these numbers have been made visual in figure 4.2. The experiment was eventually done by 42 participants, where 52,0% were female, and 48,0% were male. Only 20% were 30 years and older. This information was deducted from the survey question about gender.

Around 20 stated that they had never applied the logic theory before. The remaining entrants had either applied it and did not understand it, or applied it and understood the theory somewhat to very well. The average overall score was remarkable, where every test group was taken into account: together the participants scored 45,9% on the exercise set, where every test group had about the same amount of contributors. This is shown in figure 4.3.

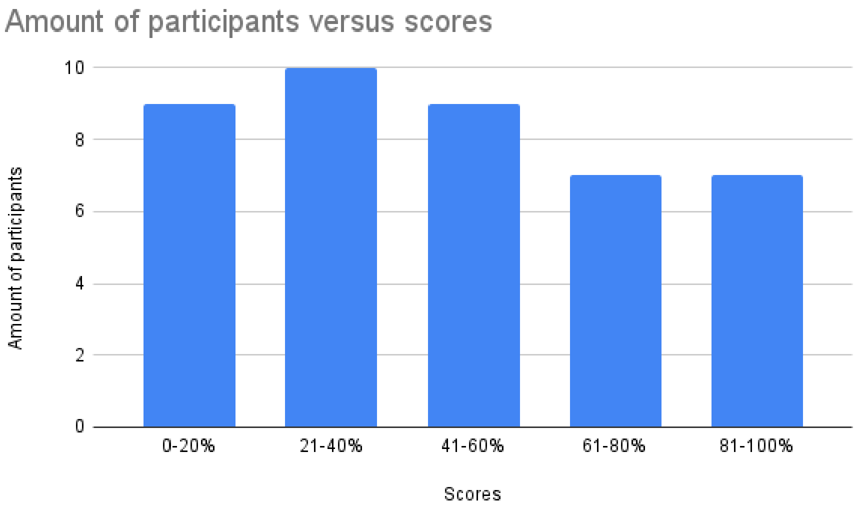
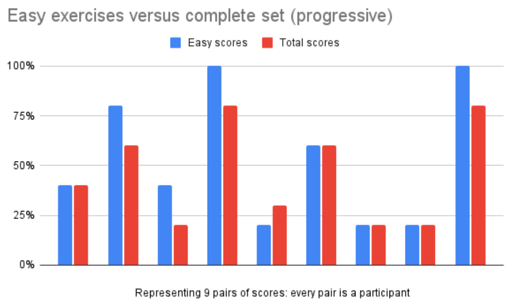
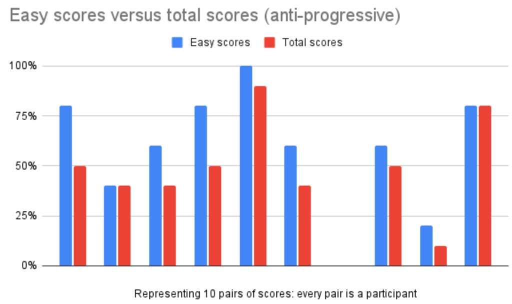


Figure 4.3: Amount of participants versus Total score

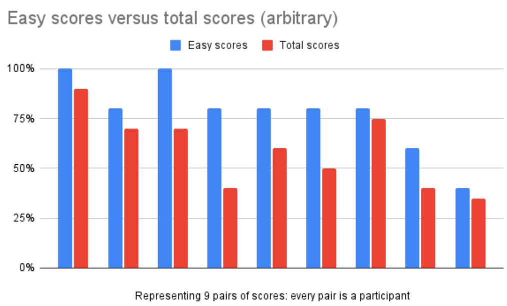


(a) Progressive: total score vs. score easiest exercises

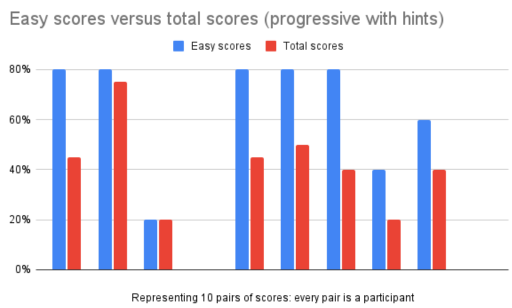


(b) Anti-progressive: total score vs. score easiest exercises

Figure 4.4: Scores



(a) Arbitrary: total score vs. score easiest exercises



(b) Progressive with hints: total score vs. score easiest exercises

Figure 4.5: Scores

4.5 Participants without a score

In total, an amount of 19 people (outside of the 42 participants) made an account at SOWISO, without actually doing the exercise set. Half of these participants underestimated the time that the experiment would take to complete. The other half did not give a reason for not completing the exercise set. Some of these contributors have logged in more than once, but could not find the exercise page when they tried to do the experiment on their mobile phones. The interface of SOWISO is affected, and changes in such a way, that it is less clear where to navigate.

4.6 Correlation 1: based on easy exercises

The first correlation is based on the easy exercises, since every exercise set is based on five exercises that vary from super easy to easy, respectively. Due to its progressive label, these were the first five exercises in group Alfa. On the other hand, group Beta ended with these five exercises, as this set is anti-progressive. Group Delta maintained the order of group Alfa, and group Gamma maintained an arbitrary order. Hence, these five exercises were interesting to focus on, based on the fact that most of the participants completed at least one of these exercises successfully. Figure 4.4 and 4.5 show that, as expected, every group scored higher on the five easiest exercises than on the five more difficult exercises. However, analysing this diagram a bit more closely, shows that every test group other than Alfa had a better score on the easiest exercises, which follows the classic learning format.

Another correlation that is based on the five easiest exercises includes the level of knowledge, which is asked in one of the survey questions. If there is a linear correlation, this could confirm that the exercises were consistent, and that the participants had given a realistic answers. It may also validate that the survey question was asked in a relevant and precise way. Figure 4.6 shows in a scatter plot that there is somewhat a linear correlation with the level of knowledge on the horizontal, and the score (on the easy exercises) on the vertical ax.

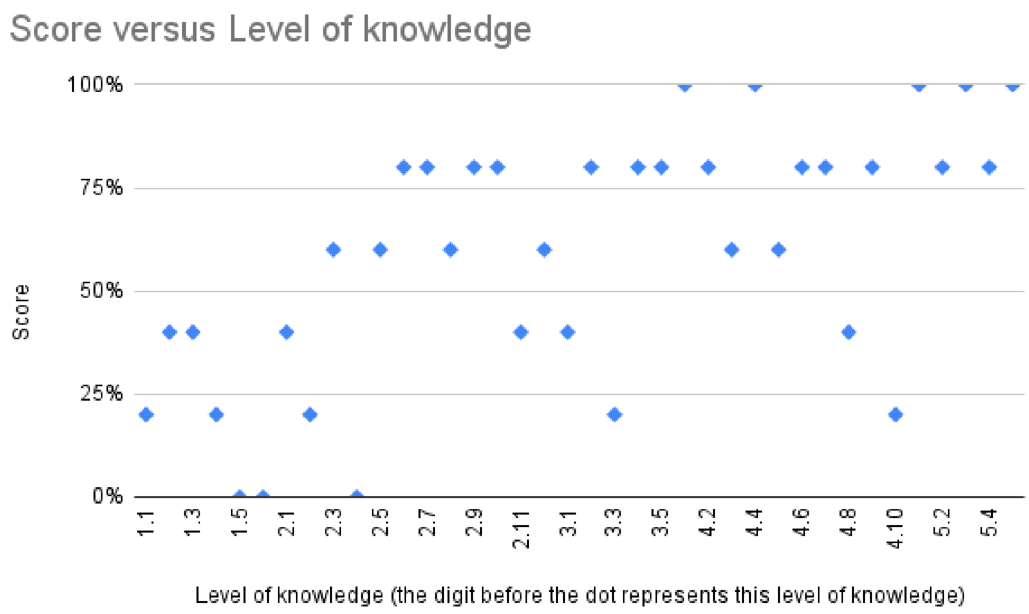
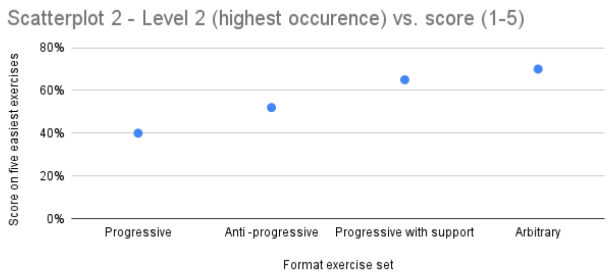


Figure 4.6: Level of knowledge vs. score on easy exercises

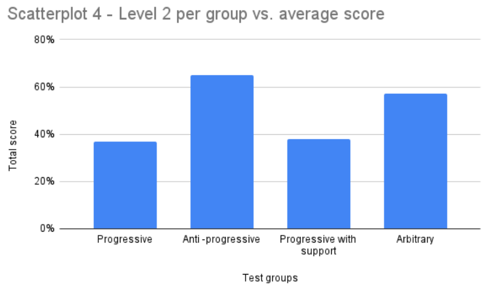
4.7 Correlation 2: based on knowledge level 2

"Knowledge level 2" is no other than a simple term that covers the group who answered the prior knowledge question from the survey. No scientific background is there to be found behind this definition. However, the survey question was stated as follows: "I have heard of, but never applied logic theory." This group will be analysed, since this was the most common answer among the participants. Every test group contains about the same amount of people with the previously stated answer to the prior knowledge question. The first correlation that is based on this knowledge level is the link to their score on (once again) the five easiest exercises. Group Alfa scored lower than any other test group, whereas group Gamma (arbitrary order) seemed to win the race.

Furthermore, the total score of this test group has a correlation in the sense that, again, group Alfa with its classic approach, scored lower than any other test group (see figure 4.8). Group Alfa (progressive) and group Delta (progressive with support) however, barely differ from each other in this diagram. This could indicate that the supportive learning factor was not as efficient as the other test groups; the hints may not have been useful enough. What is striking, is that the total score of group Delta is relatively low compared to their score on the five easiest exercises. It could be the case that the hints were only useful for the easy exercises.



(a) Level 2 (highest occurrence) vs. score on easy exercises



(b) Level 2 per group vs. average total score

Figure 4.7: Correlation based on knowledge level 2

4.8 Correlation 3: based on the creator's implementation

Another correlation to highlight is the relation between the implementation and the format of an exercise. An exercise consists of a formula, where the student must draw a corresponding model. Such a formula could correspond to one or more models. A model consists of, for example, a connection from node A to node B, and a connection from node B to itself. The more combinations of connections a formula allows, the more combinations should be included in the JavaScript code. However, did we derive a point where the amount of combinations has a different effect on the time needed per implementation? Looking at figure 4.9 and 4.10, it seems to change direction at the stage of ten possible solutions. This confirms that the steep trend in the diagram becomes less steep. This shows that the amount of possible solutions has less impact on the required implementation time.

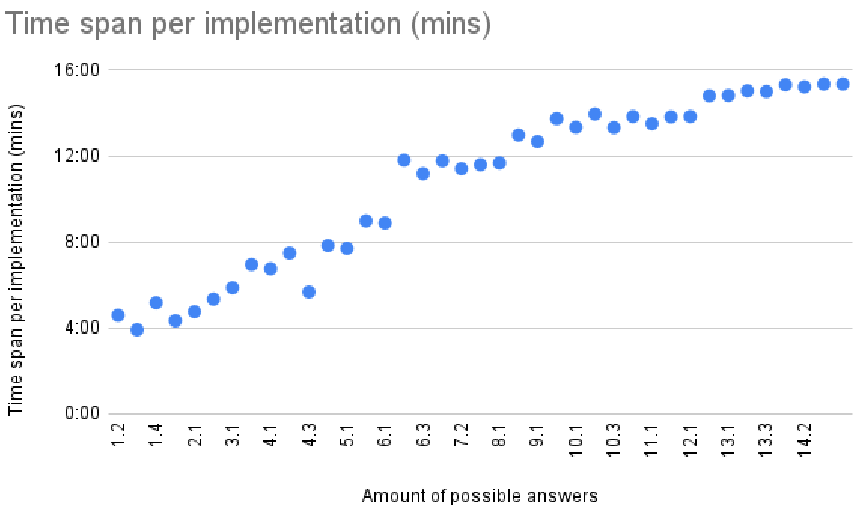
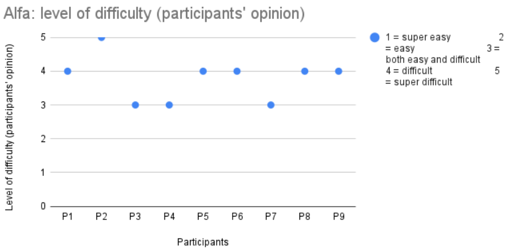
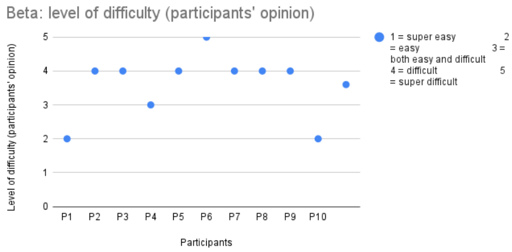


Figure 4.8: Time per implementation

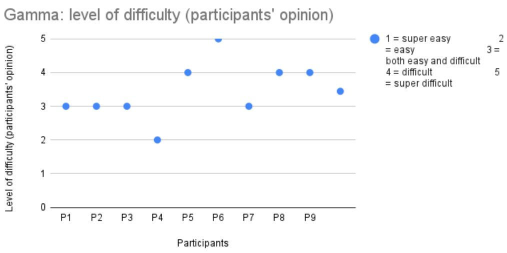


(a) Level of difficulty in group Alfa

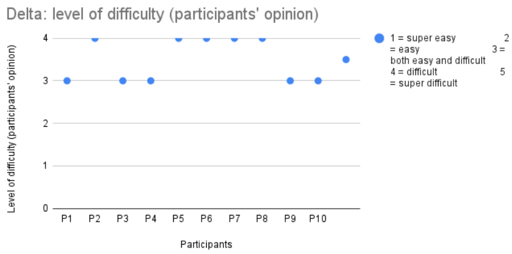


(b) Level of difficulty in group Beta

Figure 4.9: Participants' opinion



(a) Level of difficulty in group Gamma



(b) Level of difficulty in group Delta

Figure 4.10: Participants' opinion

4.9 Participants' experiences and opinion

At the end of the exercise set a question in the survey asked participants how they had experienced the test. This has been measured on a scale from one to five, where one represents super easy and five represents super difficult. [26] The overall average difficulty rate was a 3.5, meaning that the contributors found the experiment somewhat easy yet difficult. Interesting to see, is that the participants in group Alfa thought that the experiment was more difficult than all the other test groups. Groups Beta, Gamma and Delta are almost on the same line, meaning that they experienced it almost in the same way. Besides the difficulty of this experiment, most of the participants gave positive feedback, consisting the fact that they liked interactive digital exercises more than studying from books.

4.10 Significance

The previously stated results, originate from different numeric series retrieved from the experiment. The scatter plots have shown some interesting correlations. However, up until now, the significance of these numbers has been ignored. This is, because there are many factors unconsidered in the four test groups. Examples of these factors, are the amount of time they took to read the theory page, and the extra sources that they have used to get their score. Hence, I do not prefer the statistics to validate the experiment to a certain extent. However, it is an addition to the research to conduct the *p-values* of the scatter plots from figures 4.4 - 4.6. Figures 4.4 and 4.5 compares the total scores and the scores on the easiest exercises; it's *p-value = 0,095.* This value shows us, unfortunately, that the test is not significant. The other scatter plot on the other hand, contains the level of knowledge and the participants' scores on easy exercises. Surprisingly, this test ended up being significant, due to its *p-value = 0,00056.*

4.11 Discussion

4.11.1 SOWISO

Integration became the start of the method for this research, whereas the plan did not always go as expected. Understanding the platform took longer than expected as there seemed to be so many different approaches that should have been mastered before creating model designs. Trial and error was the most efficient approach to come across as many scenarios as possible. Moreover, feeling confident combining the different languages was relatively less complex, as the author's background already consists of programming skills.

Once deeper functionalities are required to complete a project, GeoGebra becomes complex. This results in the design of interactive models in regard to this research, where requirements were applied. Examples of these detailed functionalities apply when a student should only be able to modify the checkboxes, rather than the other elements displayed in and around the model. These features must be defined in JavaScript, as GeoGebra has limited options on a web application level.

Coding in JavaScript is something that everyone could learn in a short amount of time since there are a lot of built-in functions, with a straightforward application. The link between GeoGebra and JavaScript, on the other hand, covers some complex concepts. GeoGebra sometimes fails in immediately read the code implemented from a JavaScript source. The cause of this deficiency is still unknown. Every time a new model design code got loaded into the GeoGebra web application, all the current elements had to be deleted, the file had to be saved and the page should be reloaded. Once the page got refreshed, it had to be saved again, reloaded again, and then the desired updated model appears. These steps must all be repeated, every time an updated code got imported. Such a deficiency in functionalities could be a valid justification not to call this implementation efficient from an author's perspective.

4.11.2 Creator

The purpose of this research from a teacher's vision would be the absolute ease of the implementation, and the effect it should have on students' possibilities in practising exercises. However, it appeared impossible for such exercises to be adjusted on an unlimited basis. The cause of this limitation (with only one set of variables) is mostly dependent on the relation between GeoGebra and SOWISO. Once JavaScript code becomes too complex, GeoGebra does not display the variables correctly, while SOWISO can read the variables inside code. Yet, SOWISO does not allow the author of the exercise to suggest variables that are included in the implemented code in GeoGebra, as in, this connection does not exist. Therefore it is not possible to create unlimited varieties of one exercise, in the way that these models have been designed by myself and SOWISO. Still, this way of merging JavaScript, GeoGebra and SOWISO showed that it would be possible to create such an unlimited version in a period longer than six months. Besides the difficulties and the limitations that have occurred during the implementation, it is interesting to analyse the efficiency in the latest circumstances. The results displayed in figure 4.9 and 4.10 show the amount of time needed for an author to implement one exercise, depending on its answer combinations. To what extent is this contribution worth to create such an exercise? This time management only counts for creators that have been completely integrated into all the corresponding systems. If this is the case, then it becomes interesting to face options like copying and pasting implementations with the same amount of answer possibilities. This could improve the correlation between the amount of exercises and the time spent on their implementation. Copying, pasting and changing the variables in a code can be more efficient than creating a new exercise with a new amount of possible answer combinations.

Another factor that has to be taken into account, is the derivation of model answers. The first group definitely consumes more time than the second group of exercises, simply because the numbers of the models are already known. If an author succeeds in writing down the properties of a model derived by the JavaScript file, then such a model only has to be looked up once. After the first part of inefficient implementations, it can become more efficient.

4.11.3 Experiment

The experiment showed several interesting correlations. Even more compelling could be the analysis of these correlations; why do they exist and what has a significant impact on their outcomes? The first correlation was based on the easy exercises, and showed us that participants scored on average higher than on the complete exercise set. This sounds logical in general, however, looking at the value of each score, this actually tells us something. The difference between the percentage on the total score and the easy exercises is relatively small in the progressive group Alfa. While on the other hand, the score on the easy exercises is almost the lowest in group Alfa, compared to the other test groups. This explains that the progression in the anti-progressive group Beta is bigger than in Alfa. Group Beta namely starts with difficult exercises, on which their score is almost the same as the scores in Group Alfa. The scores on the easier exercises, however, are significantly higher in group Beta. Hence, the students in group Beta made more progress than in group Alfa, due to differences in results between their total score and the score on the easy exercises.

Other interesting elements occurred in group Gamma and Delta. In group Delta the score is relatively low; their format is the same as group Alfa, but then with accessible hints. How is it then possible that the scores are nearly the same, if not lower than in group Alfa? This may be the consequence of the hints either being helpful or confusing. The arbitrary group Gamma, on the other hand, resulted in a surprisingly high score on the easiest exercises, as well as the whole set. Does this then suggest that the whole order of exercises does not matter, or is there a hidden order that accidentally occurred, even though the order was derived by a random generator? It seemed to be the case that the set started with medium-hard exercises, followed by two medium-easy exercises. The medium-easy exercises were followed by a difficult, medium-easy, to easy, to super easy exercises. This could be a working pattern for an efficient learning curve.

Beside the test groups in general, there seemed to be a correlation between their level of knowledge and their score. This is an easy correlation as it is linear. The survey questions must been have both relatable as well as answered in a fair and credible way. And so it gets confirmed that the higher the level of knowledge, the higher the score on the five easiest exercises.

The most interesting correlation is based on level 2 of knowledge, meaning that a participant has heard of logic theory but never applied it before. This group scored the lowest on the five easiest exercises in the progressive group Alfa. Level 2 of knowledge is often the status of a student when starting a mathematical course. How could it be that the classic approach then fails the most? The same analysis can be done when measuring the score on the complete exercise set: the progressive group scored the lowest, while the opposite of the classic approach scored the highest. This could be caused by the fact that a participant needs time to get into the functionalities and theory of such an exercise set, meaning that the first assignment is most-likely to be failed if the participant is not yet focused on the assignment. This assumption may be confirmed by the fact that students score higher on the first easy exercises when they occur in the final phase of the set; defined as anti-progressive.

4.12 Conclusion

It was very interesting to analyse the differences between the four test groups. Even though the results showed that the progressive format did not once result in better outcomes than the other test groups, it is not fair to call the progressive approach less functioning. To answer the first research question: what would be the best format for students be to practice first order logic? I would conclude that on such a small scale, there is no such thing as a winning format. The results did not differ much from group to group. However, the other formats, anti-progressive, arbitrary, and progressive with hints, cannot be ignored. Simply, because they have not scored significantly worse than the classic approach, which is progressive.

The second research question concerned the main challenges for authors to create an undergraduate level first order logic course. The effort of an author to create a good exercise set is significant and therefore more than expected. The process of creating an exercise set is somewhat complex, as it consists of different components. The links between these components often cause the process to slow down. Errors in linking different tools cause authors to execute other tasks than authoring tasks, such as finding the malfunction or creating a detour to avoid the error. This means that an author is expected to master the several tools that are involved in the authoring process. Learning how to use the platform correctly, would take approximately 2-3 months. However, the effort might be worth it after all; students tend to enjoy the digital interactive exercises. Most of them came in without any prior knowledge and yet completed the exercise set with some new knowledge on logic content. It seems to be effective to provide students with digital learning systems, as they prefer such a format over books.

The expectations of creating an exercise concerned the feature that an exercise set could contain unlimited variables. This feature would provide students boundless opportunities to practice on the same set of exercises, with different elements. Unfortunately, I did not succeed in implementing this feature in SOWISO yet, due to a lack of time and information. It is important that such a property is added to the already existing tool; this will unburden both authors and teachers, as they are now expected to manually implement every variation of a certain exercise.

The third research question concerned the features of digital education platforms that are important for creating undergraduate-level first order logic material. In order to create a complete course, the tools, such as software, should be up-to-date and have the ability to collaborate correctly with other relevant applications. Furthermore, I would require these platforms to be easily adjustable, rather than performing the whole implementation process after every minor code-update.

Finally, I would state that every all four of the presented digital learning formats were effective to a certain extent. Such a learning system should be included in a STEM course, once it has been completely developed. This entails: the implementation of unlimited exercise variables, a correct collaboration between all the relevant applications and an easily access system that allows updates not to be time-consuming.

Bibliography

[1] C. Mercader and J. Gairín, "University teachers' perception of barriers to the use of digital technologies: the importance of the academic discipline," *International Journal of Educational Technology in Higher Education*, vol. 17, no. 1, pp. 1–14, 2020.

[2] S. V. Otterloo, "Skills for ai," 2021.

[3] D. C. Montgomery and G. C. Runger, "Applied statistics and probability for engineers," 1994.

[4] S. Kurgalin and S. Borzunov, *The discrete math workbook.* Springer, 2020.

[5] D. Matthews, "Craft beautiful equations in word with latex," *Nature*, vol. 570, no. 7760, pp. 263–265, 2019.

[6] E. M. Fedriani and R. Moyano, "Using maxima in the mathematics classroom," *International Journal for Technology in Mathematics Education*, vol. 18, no. 4, pp. 171–176, 2011.

[7] D. Zandvliet, "Towards effective learning analytics for higher education: returning meaningful dashboards to teachers.," 2020.

[8] B. A. Schwendimann, M. J. Rodriguez-Triana, A. Vozniuk, L. P. Prieto, M. S. Boroujeni, A. Holzer, D. Gillet, and P. Dillenbourg, "Perceiving learning at a glance: A systematic literature review of learning dashboard research," *IEEE Transactions on Learning Technologies*, vol. 10, no. 1, pp. 30–41, 2016.

[9] G. Veletsianos, "Digital learning environments," *Wiley Handbook of Learning Technology*, p. 242, 2016.

[10] Wikipedia, *Wikipedia.* PediaPress, 2004.

[11] M. J. Sousa, R. Cruz, J. M. Martins, *et al.*, "Digital learning methodologies and tools–a literature review," *Edulearn17 Proceedings*, pp. 5185–5192, 2017.

[12] A. Heck, "Using sowiso to realize interactive mathematical documents for learning, practising, and assessing mathematics.," *MSOR Connections*, vol. 15, no. 2, 2016.

[13] P. Kuna and M. Turčáni, "Design of new methods of teaching the subject discrete math," in *2013 IEEE 11th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pp. 247–251, IEEE, 2013.

[14] D. Gross, E. S. Pietri, G. Anderson, K. Moyano-Camihort, and M. J. Graham, "Increased preclass preparation underlies student outcome improvement in the flipped classroom," *CBE—Life Sciences Education*, vol. 14, no. 4, p. ar36, 2015.

[15] P. Strelan, A. Osborn, and E. Palmer, "The flipped classroom: A meta-analysis of effects on student performance across disciplines and education levels," *Educational Research Review*, vol. 30, p. 100314, 2020.

[16] Z. Sun, K. Xie, and L. H. Anderman, "The role of self-regulated learning in students' success in flipped undergraduate math courses (learning in groups)," *The internet and higher education*, vol. 36, pp. 41–53, 2018.

[17] D. Dicheva, B. Guy, V. Yorgov, C. Dichev, K. Irwin, and C. Mickle, "A study of using virtual currency in a discrete mathematics course," in *2021 IEEE Global Engineering Education Conference (EDUCON)*, pp. 567–576, IEEE, 2021.

[18] E. Gaudioso, M. Montero, and F. Hernandez-del Olmo, "Supporting teachers in adaptive educational systems through predictive models: A proof of concept," *Expert Systems with Applications*, vol. 39, no. 1, pp. 621–625, 2012.

[19] M. Liao, J. Patton, R. Yan, and H. Jiao, "Mining process data to detect aberrant test takers," *Measurement: Interdisciplinary Research and Perspectives*, vol. 19, no. 2, pp. 93–105, 2021.

[20] B. Martin, A. Mitrovic, K. R. Koedinger, and S. Mathan, "Evaluating and improving adaptive educational systems with learning curves," *User Modeling and User-Adapted Interaction*, vol. 21, no. 3, pp. 249–283, 2011.

[21] E. Matusov, "Progressive education is the opium of the educators," *Integrative Psychological and Behavioral Science*, pp. 1–34, 2021.

[22] C. Sangwin, "A brief review of geogebra: dynamic mathematics," *MSor Connections*, vol. 7, no. 2, p. 36, 2007.

[23] A. Guha, C. Saftoiu, and S. Krishnamurthi, "The essence of javascript," in *European conference on Object-oriented programming*, pp. 126–150, Springer, 2010.

[24] L. Shao, T. Schleicher, M. Behrisch, T. Schreck, I. Sipiran, and D. A. Keim, "Guiding the exploration of scatter plot data using motif-based interest measures," *Journal of Visual Languages & Computing*, vol. 36, pp. 1–12, 2016.

[25] J. Shrestha, "P-value: A true test of significance in agricultural research," 2019.

[26] H. Taherdoost, "What is the best response scale for survey and questionnaire design; review of different lengths of rating scale/attitude scale/likert scale," *Hamed Taherdoost*, pp. 1–10, 2019.

Appendix A

SOWISO results

Student	Student nu	groups	Summary		Alfa	
			Score	Progress	Score	Progress
[redacted]	[redacted]	[redacted]	60	100	60	100
[redacted]	[redacted]	[redacted]	83	100	83	100
[redacted]	[redacted]	[redacted]	96	94	96	94
[redacted]	[redacted]	[redacted]	93	100	93	100
[redacted]	[redacted]	[redacted]	48	100	48	100
[redacted]	[redacted]	[redacted]	48	100	48	100
[redacted]	[redacted]	[redacted]	80	94	80	94
[redacted]	[redacted]	[redacted]	33	19	33	19
[redacted]	[redacted]	[redacted]	73	94	73	94
[redacted]	[redacted]	[redacted]	46	100	46	100
[redacted]	[redacted]	[redacted]	84	100	84	100
[redacted]	[redacted]	[redacted]	53	100	53	100

Figure A.1: Formulas: Alfa, Beta, Delta and Gamma

Student	Student nu	groups	Summary		Bèta	
			Score	Progress	Score	Progress
[redacted]	[redacted]	[redacted]	66	100	66	100
[redacted]	[redacted]	[redacted]	44	100	44	100
[redacted]	[redacted]	[redacted]	68	100	68	100
[redacted]	[redacted]	[redacted]	90	100	90	100
[redacted]	[redacted]	[redacted]	63	100	63	100
[redacted]	[redacted]	[redacted]	32	100	32	100
[redacted]	[redacted]	[redacted]	67	100	67	100
[redacted]	[redacted]	[redacted]	41	100	41	100
[redacted]	[redacted]	[redacted]	84	100	84	100

Figure A.2: Formulas: Alfa, Beta, Delta and Gamma

Student	Student nu	groups	Summary Score	Progress	Delta Score	Progress
[redacted]			33	19	33	19
[redacted]			68	100	68	100
[redacted]			83	100	83	100
[redacted]			100	6	100	6
[redacted]			46	100	46	100
[redacted]			26	100	26	100
[redacted]			69	100	69	100
[redacted]			70	100	70	100
[redacted]			64	100	64	100
[redacted]			44	100	44	100
[redacted]			84	100	84	100
[redacted]			32	100	32	100
[redacted]			72	63	72	63

Figure A.3: Formulas: Alfa, Beta, Delta and Gamma

Student	Student nu	groups	Summary Score	Progress	Gamma Score	Progress
[redacted]			94	100	94	100
[redacted]			88	100	88	100
[redacted]			80	100	80	100
[redacted]			60	100	60	100
[redacted]			73	100	73	100
[redacted]			67	100	67	100
[redacted]			83	100	83	100
[redacted]			60	100	60	100
[redacted]			63	100	63	100

Figure A.4: Formulas: Alfa, Beta, Delta and Gamma

Appendix B

Exercise format

B.1 Exercise order per test group

Alfa/Delta
Which ones could be included in a propositional logic statement? ∧ ! ¬ + ∨ - >/
Big(a)
¬Linked(a,b)
∃x Linked(x,a) ∧ Linked(b,b)
∃x (Linked(x,b) ∧ ¬Linked(b,x))
∃ x ∀ y ¬Linked(x,y))
∀ x ∀ y Small(x) → Linked(y,x)
∃ x ∀ y Linked(x,y) ∧ ∃ z Linked(z,b)
∀ x Linked(x,b) ∧ (∀ xLinked(x,x) ∧ ∃ yLinked(y,y))
∀ x ∀ y (¬ Linked(x,y) → (Linked(y,x) ∨ (Linked(x,x) ∧ ¬ Linked(y,y))) ∧ ¬ Linked(a,b))

Beta
∀ x ∀ y (¬ Linked(x,y) → (Linked(y,x) ∨ (Linked(x,x) ∧ ¬ Linked(y,y))) ∧ ¬ Linked(a,b))
∀ x Linked(x,b) ∧ (∀ xLinked(x,x) ∧ ∃ yLinked(y,y))
∃ x ∀ y Linked(x,y) ∧ ∃ z Linked(z,b)
∀ x ∀ y Small(x) → Linked(y,x)
∃ x ∀ y ¬Linked(x,y))
∃x (Linked(x,b) ∧ ¬Linked(b,x))
∃x Linked(x,a) ∧ Linked(b,b)
¬Linked(a,b)
Big(a)
Which ones could be included in a propositional logic statement? ∧ ! ¬ + ∨ - >/

Gamma
∃ x ∀ y Linked(x,y) ∧ ∃ z Linked(z,b)
∀ x Linked(x,b) ∧ (∀ xLinked(x,x) ∧ ∃ yLinked(y,y))
∃x Linked(x,a) ∧ Linked(b,b)
Which ones could be included in a propositional logic statement? ∧ ! ¬ + ∨ - >/
∃x (Linked(x,b) ∧ ¬Linked(b,x))
∃ x ∀ y ¬Linked(x,y))
¬Linked(a,b)
∀ x ∀ y Small(x) → Linked(y,x)
Big(a)
∀ x ∀ y (¬ Linked(x,y) → (Linked(y,x) ∨ (Linked(x,x) ∧ ¬ Linked(y,y))) ∧ ¬ Linked(a,b))

Figure B.1: Formulas: Alfa, Beta, Delta and Gamma

B.2 Post-test evaluation questions

Complete exercise set: Difficulty assignments (Exercise id: 87821)

How difficult was this set of exercises to you on a scale from 1-5?

- Super easy
- Easy
- Somewhat easy, somewhat difficult
- Difficult
- Very difficult

Check

Solution

Complete exercise set (Exercise id: 87978)

Do you have any suggestions on how to improve this exercise set? After answering this question, you can give your experience a grade.

Think for example of the length, difficulty or the information provision etc.

Solution

Complete exercise set: Education (Exercise id: 87789)

What is your highest education completed?

Check

Solution

Complete exercise set: Prior knowledge (Exercise id: 87817)

To what extent did you have prior knowledge on Logic theory?

- I had never heard of Logic theory.
- I have heard of, but never applied Logic theory
- I have applied Logic theory before, but did not understand it.
- I have applied Logic theory before, and I somewhat understood theory.
- I have applied Logic theory, and I could follow almost everything of the theory.
- I have applied Logic theory, and I think I would be able to teach the course.
- Other

Check

Solution

Complete exercise set: Gender (Exercise id: 87787)

What is your gender?

- Female
- Male
- Other

Check

Solution

Complete exercise set: Age (Exercise id: 87788)

How old are you right now?

Check

Solution

Appendix C

JavaScript (defining models)

```
/* This is the code for the model:and(forall(y, linked(b, y)), not(exists(x, linked(a, x))))
  The format of this code has been used to create every model in GeoGebra.*/

function ggbOnInit(){
 ggbApplet.registerAddListener("swStyle");
 ggbApplet.evalCommand("swCorrect = false");
 ggbApplet.evalCommand("A = (-2,1)");
 ggbApplet.evalCommand("B = (1.8,-1.7)");
 ggbApplet.evalCommand("C = (-2,1)");
 ggbApplet.evalCommand("D = (-2.5,1.34)");
 ggbApplet.evalCommand("E = (-1.89,1.09)");
 ggbApplet.evalCommand("F = (-2.03187,1.46079)");
 ggbApplet.evalCommand("G = (1.32,-1.34)");
 ggbApplet.evalCommand("H = (1.8,-1.57)");
 ggbApplet.evalCommand("I = (-2.29355,0.73307)");
 ggbApplet.evalCommand("J = (-1.42557,0.98074)");
 ggbApplet.evalCommand("K = (1.70439,-1.29444)");
 ggbApplet.evalCommand("L = (1.74646,-1.94446)");
 ggbApplet.evalCommand("M = (2.33766,-1.44692)");

 ggbApplet.evalCommand("LB = (-3,-2)");
 ggbApplet.evalCommand("RB = (3,-2)");
 ggbApplet.evalCommand("RT = (3,2)");
 ggbApplet.evalCommand("LT = (-3,2)");

 ggbApplet.evalCommand("ab = Vector(A,B)");
 ggbApplet.evalCommand("ba = Vector(B,A)");
 ggbApplet.evalCommand("AAA = Vector(C,E)");
 ggbApplet.evalCommand("BBB = Vector(B,H)");
 ggbApplet.evalCommand("aa = CircumcircularArc(C,D,E)");
 ggbApplet.evalCommand("bb = CircumcircularArc(B,G,H)");
 ggbApplet.evalCommand("ABig = Circle(F,I,J)");
 ggbApplet.evalCommand("BBig = Circle(K,L,M)");

 ggbApplet.evalCommand("AA = Checkbox({AAA,aa})");
 ggbApplet.evalCommand("AB = Checkbox({ab})");
 ggbApplet.evalCommand("BA = Checkbox({ba})");
 ggbApplet.evalCommand("BB = Checkbox({BBB,bb})");
 ggbApplet.evalCommand("BigA = Checkbox({ABig})");
 ggbApplet.evalCommand("BigB = Checkbox({BBig})");

 ggbApplet.evalCommand("universe = Polygon(LB,RB,RT,LT)");
 swStyle2();
```

```
  ggbApplet.registerObjectUpdateListener("AA","swCorrect");
  ggbApplet.registerObjectUpdateListener("AB","swCorrect");
  ggbApplet.registerObjectUpdateListener("BA","swCorrect");
  ggbApplet.registerObjectUpdateListener("BB","swCorrect");
  ggbApplet.registerObjectUpdateListener("BigA","swCorrect");
  ggbApplet.registerObjectUpdateListener("BigB","swCorrect");
}

function swCorrect(){
  if ((ggbApplet.getValue("BA") && ggbApplet.getValue("BB")) == 1 &&
(ggbApplet.getValue("AA") && ggbApplet.getValue("AB") && ggbApplet.getValue("BigA") &&
ggbApplet.getValue("BigB")) == 0) {
    ggbApplet.evalCommand("swCorrect = true");
  }
  else if ((ggbApplet.getValue("BA") && ggbApplet.getValue("BB") &&
ggbApplet.getValue("BigA")) == 1 && (ggbApplet.getValue("AA") &&
ggbApplet.getValue("AB") && ggbApplet.getValue("BigB")) == 0) {
      ggbApplet.evalCommand("swCorrect = true");
  }
  else if ((ggbApplet.getValue("BA") && ggbApplet.getValue("BB") &&
ggbApplet.getValue("BigB")) == 1 && (ggbApplet.getValue("AA") &&
ggbApplet.getValue("AB") && ggbApplet.getValue("BigA")) == 0) {
      ggbApplet.evalCommand("swCorrect = true");
  }
  else if ((ggbApplet.getValue("BA") && ggbApplet.getValue("BB") &&
ggbApplet.getValue("BigA") && ggbApplet.getValue("BigB")) == 1 &&
(ggbApplet.getValue("AA") && ggbApplet.getValue("AB")) == 0) {
      ggbApplet.evalCommand("swCorrect = true");
  }
  else {
    ggbApplet.evalCommand("swCorrect = false");
  }
}

function swStyle(object){
  var type = ggbApplet.getObjectType(object);
  if(type != "boolean"){
    ggbApplet.setLabelVisible(object, false);
  }

  if(type == "point"){
    if(object == "A" || object == "B"){
      ggbApplet.setPointSize(object, 3);
```

```
      ggbApplet.setPointStyle(object, 10);
      ggbApplet.setColor(object,0,0,128);
      ggbApplet.setLabelVisible(object, true);
      ggbApplet.setFixed(object, true, true);
    }
    else{
      ggbApplet.setVisible(object, false);
    }
  }
  else if(type == "boolean"){
    ggbApplet.setValue(object, false);
  }

  else if(type == "vector" || type == "CircumcircularArc" || type == "Circle"){
    ggbApplet.setVisible(object, false);
  }
}

function swStyle2(){
  ggbApplet.setFilling("universe", 0);
  ggbApplet.setColor("universe", 0, 0, 0);
  ggbApplet.setFixed("universe", true, false);
  for(i = 0; i < ggbApplet.getAllObjectNames(["segment"]); i++){
    console.log(ggbApplet.getAllObjectNames(["segment"])[i]);
    ggbApplet.setLineThickness(ggbApplet.getAllObjectNames(["segment"])[i], 3);
    ggbApplet.setColor(ggbApplet.getAllObjectNames(["segment"])[i], 0, 0, 0);
    ggbApplet.setFixed(ggbApplet.getAllObjectNames(["segment"])[i], true, false);
  }
}
```