AIMS Mathematics, 9(3): 5234–5249.
DOI: 10.3934/math.2024253
Received: 27 October 2023
Revised: 01 January 2024
Accepted: 05 January 2024
Published: 26 January 2024

http://www.aimspress.com/journal/Math

Research article

Problems and challenges of using randomized automatically evaluating geometric construction problems in Moodle LMS

Peter Csiba*and Peter Vajo

Department of Mathematics, J. Selye University, Bratislavská cesta 3322, Komárno, 945 01, Slovakia

* **Correspondence:** Email: csibap@ujs.sk.

Abstract: The objective of this paper was to examine the present state of geometric constructions in both international and national mathematics assessments. Additionally, it aimed to shed light on the noticeable lack of geometric construction items in these assessments and propose a solution to incorporate such items by utilizing a web-based mathematics assessment tool. After providing a brief summary of the unique benefits associated with geometric construction problems, the subsequent paper explored the development of a web-based mathematics assessment tool, the procedure for generating automatic evaluation GeoGebra applets for geometric construction problems, and their integration into Moodle learning management system (LMS). The purpose of this assessment tool was to enhance the evaluation of a substantially larger group of students within the realm of geometric constructions. The final section of the paper examined the difficulties encountered in the creation of the automatic evaluation applets and their incorporation into Moodle LMS. Through the development of such a web-based mathematics assessment tool, we enabled the evaluation of a significantly larger number of students, thus aiding not only national and international mathematics examinations but also assisting mathematics teachers in tracking students' scores and progress.

Keywords: mathematics assessment; geometric constructions; automatic evaluation; assessment tool; difficulties and solutions
Mathematics Subject Classification: 97D60, 97G99, 97U50, 97B50

1. Introduction

There are a number of well-known international (TIMSS, PISA-see subsection 1.1) and national studies and assessments that evaluate the mathematical performance of students. In the Slovak Republic, the most important national mathematics examinations are Testing 5 (https://www2.nucem.sk/sk/merania/narodne-merania/testovanie-5/roky/2022-2023), Testing 9 (https://www2.nucem.sk/sk/merania/narodne-merania/testovanie-9), and Maturita (school-leaving

examination) (https://www2.nucem.sk/sk/merania/narodne-merania/maturita).

As part of a pilot program, the tests and exams were administered exclusively online in some institutions, while in others they were administered on paper but evaluated by machine tools (scanned and interpreted by image recognition software). This assured both the swift evaluation of the tests and the objectivity and impartiality of the evaluation. However, the technical validation bottleneck only permits the selection of test options (a, b, c, and d) and fill-in tasks that can be expressed as numbers (e.g., adjusted to two decimals).

Although the tests were created to ensure that all areas of literacy, including geometry (geometry problems in these tests are typically computational problems with a numerical solution), are covered, they do not include geometric construction problems, which cannot be evaluated in this manner.

1.1. International mathematics assessments

PISA (https://www.oecd.org/pisa/) is an internationally conducted study designed to assess the educational systems of various countries by examining the abilities and knowledge of 15-year-old students in participating nations. At regular intervals of three years, a sample of fifteen-year-old individuals is chosen through a random selection process to undergo examinations in fundamental academic disciplines, including reading, mathematics, and science. Since the turn of the millennium, more than 70 countries and economies have participated in PISA. Each year of assessment emphasizes one particular subject. Upon examination of a Pisa test (https://www.oecd.org/pisa/pisaproducts/pisa2012-2006-rel-items-maths-ENG.pdf) that centers on the subject of mathematics, the geometry-related questions primarily revolve around the computation of area, the utilization of the Pythagorean theorem, and the precise application of measurement techniques to determine the hypotenuse of a right triangle. Additionally, the examination assesses the ability to convert measurements on a scale drawing, compute lengths based on information presented in a two-dimensional depiction, interpret the necessary perspective from a photograph of a three-dimensional structure, calculate the central angle of a circular sector, and determine the length of an arc.

TIMSS (https://timssandpirls.bc.edu/) is a comprehensive evaluation that measures the proficiency of students in mathematics and science across different nations. It specifically targets fourth (or fifth) and eighth (or ninth) grade students, with participation from over 64 countries. The geometry-related mathematics questions in the TIMSS released items (`https://nces.ed.gov/timss/pdf/TIMSS2011_G8_Math.pdf`) encompassed several topics such as area, length, and angle computation, determination of distances between midpoints, questions regarding geometric transformations, application of the Pythagorean theorem, and the task of drawing an isosceles triangle on grid paper.

Despite the fact that both of these international tests prioritize mathematics and incorporate test items relating to geometry, it is apparent that geometric constructions are noticeably absent.

1.2. National mathematics assessments

Primary education is compulsory and lasts for nine years in the Slovak Republic. This educational level is divided into two stages: grades 1–5 (6–10-year-olds) and grades 5–9 (10–15-year-olds). After successfully completing primary school, students are eligible for secondary education. Secondary education is not compulsory, and not all types of secondary schools administer a school-leaving examination at the end of secondary education. In all primary schools, the national

examinations Testing 5 and Testing 9 (the examinations for students in the fifth and ninth grades, respectively) are used to assess students' knowledge and abilities, according to the National Institute for Certified Educational Measurements (https://www2.nucem.sk/sk/merania). Mathematics is a significant component of both the fifth-grade and the ninth-grade national examinations, Testing 5 and Testing 9. The school-leaving examination ('Maturita') is mandatory for all students of secondary grammar schools, schools of art, and vocational schools in the final year of secondary education. Maturita aims to verify and assess the knowledge and skills of secondary school graduates, and, according to its official website (https://www2.nucem.sk/en/measurements/maturita/about-maturita), tests in mathematics not only aim to verify students' knowledge and skills but also to promote mathematical literacy.

Given the absence of geometric construction problems, which constitute the basis of each grade's curriculum, the accuracy of this statement can be questioned.

From the results of each year's national examinations, we can see an apparent decline in the success rate of solving geometry problems in general. While geometry scores in Testing 5 (https://www2.nucem.sk/sk/merania/narodne-merania/testovanie-5/roky/2022-2023) do not appear to diverge from those of other mathematical topics, data from Testing 9 (https://www2.nucem.sk/sk/merania/narodne-merania/testovanie-9) indicates a negatively developing disparity between geometry and the other mathematical topics among grade nine students.

In the two most recent years of school-leaving examinations in mathematics, we saw a success rate of 54 and 53,8%, with 11,35 and 11,28% of students choosing to take the exam.

As schools are primarily concerned with their students' performance on national examinations, teachers are expected to prepare their students for optimal performance. Due to the aforementioned factors, geometric constructions are frequently relegated to the background of the curriculum as they are not included in national examinations.

1.3. The importance of geometric constructions

The systematic value of geometric constructions dates back to ancient Greek mathematicians. From Euclid's Elements [1], which devotes a great deal of time to demonstrating how to draw geometrical figures with a ruler and compass, to the three famous construction problems (doubling the cube, trisecting the angle, and squaring the circle), geometric constructions have been a subject of study for many centuries. The traditional position of constructing figures in the teaching of plane geometry is justified, according to Polya [2], by their ability to expose students to problem-solving.

Frequently, problem-solving requires creativity and the identification of a solution (or solutions) that are not immediately apparent. In geometric construction problems, there are commonly multiple correct solutions and correct approaches to solving a problem. However, the solutions are not immediately apparent; rather, students must connect geometric and mathematical concepts and ideas. Consequently, construction problems can be used to evaluate students' capacity to connect mathematical concepts. According to Levav and Leikin [3], linking mathematical ideas entails connecting new ideas to related ones and solving challenging mathematical problems by applying previously learned concepts and techniques. Even a triangle construction problem where the altitude and median are given or a pentagon construction problem requires a more profound understanding of geometry concepts.

Contrary to common misconceptions, geometric constructions are not concerned with how we hold

a compass or ruler or how precise we are. Each construction step is founded on a close logical relationship, and each locus is applied in accordance with these logical relationships and properties. A comprehensive design (Pólya) is required to account for the appropriate knowledge and relationships between the data provided in the task and the shape to be constructed. Providing a problem with general, non-concrete input data also impacts the probability of a solution and the number of solutions. Although this process uses well-known techniques and procedures, they are less algorithmic and deterministic than those that students are more likely to encounter (such as the division algorithm or the Euclidean algorithm). This provides students with a better understanding of procedures for solving mathematical problems.

In order for students' creativity to flourish and for national mathematics examinations to include geometric construction problems, a new assessment instrument is required. According to Venturini and Sinclair [4], researchers have already acknowledged the underutilized value of using dynamic geometry environments for student assessment; however, a plausible national-level dynamic geometry instrument for assessing students' performance in geometric construction is still not present.

1.4. Problems with assessing geometric constructions

When it comes to national assessments, a large number of topics can be evaluated simply by writing responses or solutions in answer boxes, which simplifies the evaluation process for both human and potentially automatic answer verification (comparison of the students' solutions to the correct solutions). Using this standard automatic evaluation method, geometry problems involving the calculation of length (of a side), perimeter, area, volume, surface area, or angle can be easily evaluated, as the solutions to each of these problems are numerical. One aspect of geometry, however, appears to be overlooked in these national assessments and tests. Geometric constructions involve problems with non-numerical solutions that are constructed with a ruler and compass and cannot be copied into answer boxes for automatic evaluation. The creation and evaluation of geometric construction problems for a larger number of participants has proven to be extremely difficult; as a result, neither Testing 5 and 9 nor the school-leaving examination contain such problems.

There have been advancements in the online space for solving mathematical exercises in a more efficient way. For online testing, there are several well-documented and parameterizable frameworks for creating problems and problem sets for many branches of mathematics, such as the WebWork [5] system supported by the Mathematical Association of America (MAA), as well as the Learning Management System (LMS), which, in addition to the standard static testing tools, can be used to create parameterizable exercises with the help of appropriate plugins (e.g., STACK - `https://moodle.org/plugins/qtype_stack`, Wiris in Moodle). As a result, each (or nearly every) student is given a unique exercise (with different values), making it extremely difficult to copy assignments and cheat on exams. Automatic scoring is another major advantage of these online tests. If the exercises are created correctly, the teacher does not need to correct the tests, and the student receives immediate feedback on whether or not he or she has correctly solved the exercise or test. Unfortunately, these tools cannot be used to create geometric construction problems.

2. One step closer with GeoGebra

GeoGebra [6] is a dynamic mathematics software for all educational levels, and it is extremely popular among mathematicians and mathematics instructors. It is frequently used to illustrate geometric concepts, constructions, and ideas, playing a crucial role as a visualization instrument that significantly improves students' comprehension and geometry achievements, as demonstrated by multiple studies [7–11].

GeoGebra is free to use as both a computer or mobile application and a web application. With just a few clicks, interactive applets created with GeoGebra can be shared publicly on the GeoGebra website, even from the desktop application. In fact, GeoGebra's website contains a Classroom section (https://www.geogebra.org/m/hncrgruu) where complex materials can be shared with students. Additionally, the toolbar can be modified so that students only have access to the tools that the instructor wants them to use.

Furthermore, GeoGebra allows users to generate problems with automatic verification. The process of creating construction problems with automatic verification is different from the creation of algebraic problems. The solver is not restricted to a single solution, as there may be multiple distinct or symmetrically valid solutions, and the automatic verification function must include all potential solutions in order to accurately verify the solver's output—a construction in our case. In addition, by utilizing GeoGebra, the minor imperfections that typically occur during ruler and compass construction are eliminated, while the underlying principles of each construction step are maintained and the process is accelerated in comparison to handcrafted construction. Teachers can create GeoGebra applets with automatic evaluation and share those with their students, or they can create a GeoGebra book or interactive mathematics textbook out of the GeoGebra applets, as seen in [12]. While this can meet the pedagogical and didactic needs of students with plenty of interactivity and feedback during the solving process, teachers are not able to track their students scores and progress.

If digital tools and digital competencies are included in national curricula, as article [13] contends, then dynamic geometry software is an underutilized resource in the modernized mathematics classroom. Research regarding the topic of creating geometric construction problems with automatic evaluation and integrating them into a learning management system that can track students progress is very limited. The GeoTest [14], developed by Gergelitsová and Holan [15], is one notable attempt to develop such an automatic evaluation system for geometric constructions in recent years. It is an automatic evaluation system that is based on GeoGebra and is mostly used in the Czech Republic. The creation of a dynamic geometry environment also piqued Dick's [16] interest. However, there was still a need for an automatic evaluation system that mathematics teachers could use, edit, and construct problems without the assistance of programmers or the mathematics teacher needing to be a programmer. In the following section, we will introduce one potential solution to this problem by presenting our free, web-accessible electronic system (https://matek.ujs.sk/) containing a collection of exercises with an automatic evaluation for grades five through nine.

3. The creation of a web-accessible mathematics assessment tool (that includes geometric constructions)

The fundamental challenge would be how to generate randomized geometric construction problems with random parameters in a web-based, online interface that evaluates and manages the students' work, provides them with immediate feedback on the correctness of their solution, and can be implemented without the need for serious programming knowledge. We consider the randomization of the given data of the tasks to be important because, on the one hand, although the nature of the task does not change, the result does and, therefore, copying the solution or solving the tasks together is no longer trivial. On the other hand, a randomized problem can be used multiple times in other tests because it generates unique data in each of them.

As a result of GeoGebra's programming, we can use JavaScript to set up automatic evaluation in applets we create; further details are provided below. Mathematics teachers may not possess programming skills; therefore, they highly appreciate the utilization of a learning management system that does not necessitate programming knowledge. Additionally, the availability of preexisting scripts is particularly desirable. Moodle LMS [17] appears to be the most rational option, given its status as the predominant free learning management system, particularly in Central Europe, and its lack of necessity for teachers to possess extensive programming expertise. As Moodle assignments, the Moodle plugin (`https://moodle.org/plugins/qtype_geogebra`) can manage GeoGebra applets with automatic evaluation. The plugin adds a new question type (GeoGebra) to Moodle, which can be selected when creating a question to specify which variables are randomized by Moodle, which conditions are set for them, and which logical variables are scored if they are true.

In recent years, we have developed a Moodle plugin for GeoGebra that allows the creation of geometric construction problems. In the following, we hope to assist interested developers and instructors by presenting our know-how, highlighting unanticipated problems encountered during development, and proposing solutions and ideas.

In the majority of schools in the Slovak Republic, Slovak is the language of instruction; however, there are also schools that teach in the languages of minorities and ethnic groups. Hungarian is an example of such a language. The website (https://matek.ujs.sk/) was designed to assist mathematics teachers in upper elementary schools where Hungarian is used as the language of instruction. It is a web-based, free electronic system comprising a collection of exercises with automatic evaluation in Hungarian.

When it comes to geometric construction problems with automatic evaluation, several important factors must be considered:

1. `The problems require a randomizing component`
   When a problem-solver opens a set of problems, points with different coordinates or segments of variable length must be presented, resulting in a unique challenge each time. In triangle construction problems, additional factors, such as triangle inequality or median (m_a) ≥ altitude (h_a) inequality, must be considered to assure the constructability of the triangle. One way to accomplish this is through the use of GeoGebra's tools, such as the Sliders and the Random (RandomBetween) commands.
   The issue is that you have multiple randomization options, but using them in a single applet poses

no problem. However, when we attempt to embed our functioning applet into Moodle with the plugin, certain randomization parameters of GeoGebra, such as random points or that an input variable depends on other input variables, are not allowed. Variables that are permitted and can be parameterized in Moodle are those that are defined with a slider in GeoGebra and whose limits are specific values. In the case of a triangle with three sides, for instance, two side lengths (a and b) can be arbitrarily specified with a slider so that the interval, which is the slider's boundary, is comprised of two positive values. To be able to construct the triangle, the triangle inequality must be satisfied for the third side (c), namely, $|a - b| \leq c \leq a + b$. We can solve this in GeoGebra by setting the boundaries of c to $(|a - b| + \varepsilon)$ and $(a + b - \varepsilon)$, where ε is a small positive number, such as $0,1$. However, the GeoGebra plugin for Mooodle does not accept or support such dynamically specified ranges.

2. `Script that allows the accurate evaluation of the provided answer`
   The GeoGebra script compares the solver's answer to the given correct answer, and if they are identical, the boolean variable is set to true and the solver is awarded a point. There are instances in which there are multiple valid solutions to geometric construction problems (e.g., a parallelogram construction). Therefore, all correct answers must be added in advance.
3. `Consider the available construction space`
   When the construction problem appears as a GeoGebra applet, there should be sufficient space to construct and display the construction.
   If the applet is embedded in a website or LMS, it must be scaled to "fit" the page and display on the majority of devices. For an applet of a given dimension, the exercise should be designed so that, for any randomized parameter setting, the given elements of the exercise are visible and the necessary elements of the construction fit within the given area. Although GeoGebra provides tools for moving and zooming in and out of the worksheet, this is not expected of the student solving the problem.

In the following chapter, we discuss the issues and solutions that arose during the development of the free, web-accessible electronic system when these important factors for generating geometric construction problems with automatic evaluation were taken into consideration.

3.1. Solutions to problems encountered when creating geometric construction problems with automatic evaluation for our free, web-accessible electronic system

First, we will discuss briefly how to create a GeoGebra applet that automatically evaluates geometric construction problems. In general, we construct the object that the problem requires the solver to construct and hide it (this will be the target object). Next, right-clicking the object and opening the setting will lead us to the Scripting option, where in the Global JavaScript field we specify a script that compares each object created during the task to the target object and creates a logical variable with a true value if a match is found.

In the simple demonstration applet (https://www.geogebra.org/m/c2u2yppq) where students have to construct the line that is perpendicular to a given line and passes through a given point, the following script (as shown in Figure 1) is used:

```
function ggbOnInit() {
    ggbApplet.debug("ggbOnInit");
    ggbApplet.registerAddListener("newObjectListener");
}

function newObjectListener(obj) {
    if (obj != "finished") {
        var cmd = "finished = (" + obj + "==target)";
        ggbApplet.debug(cmd);
        ggbApplet.evalCommand(cmd);
        finished = ggbApplet.getValueString("finished");
        if (finished.indexOf("true") > -1) {
            ggbApplet.setVisible("solution", true);
        }
    }

}
```

Figure 1. Global JavaScript for automatic evaluation.

The toolbar has been customized so that it contains only the tools required to construct the solution.

When creating an automatically evaluating randomized applet, we oftentimes depart significantly from the conventional procedure (in which we begin with the data and proceed to the solution). Consider the task of constructing the inscribed circle of a random triangle, as shown in the following applet (https://www.geogebra.org/m/ydqecdda). We would ordinarily begin by adding an arbitrary triangle. If the vertices are chosen at random (for example, using the Randompoint command on the GeoGebra command line) or if they are constructed from the sides corresponding to the triangle inequality described above, we worry that the resulting triangle will be too small or too ”sharp, blunt, flat” for construction, which could make the online construction time-consuming and error-prone. To solve this problem, we first choose the inscribed circle to be constructed as the target object; its radius is parametrizable using a slider. We select a random point on the circle, rotate it twice (or potentially three times) around the circle's center by a random angle, and select the tangents at these points to obtain a tangent triangle of the circle. We choose the boundaries of the intervals so that the tangent triangle extends as far as possible within the designated drawing area. Using this method, we can eradicate the presumption that one side of a triangle must be horizontal, as well as Problem 3 from the previous section. We can then hide everything but the triangle, including unnecessary toolbar elements, and the generated script will search for the specified circle as the target function. Obviously, this is just a simple method to solve the problem; we do not rule out the possibility of a more sophisticated solution. Naturally, the solution involves constructing the angular bisectors of the interior angles to obtain the center of the inscribed circle, followed by a tangent point for the radius using a perpendicular; however, as can be seen, these were not included at all in the creation of the construction problem.

When opening, launching, or reloading a randomized applet, the random items are always chosen at random, and the corresponding figure is displayed. The initial stages of creating a randomized problem are performed in GeoGebra (an applet), and then the Geogebra_qtype plugin must be used to embed the problem in Moodle. In several instances, we discovered that the plugin only accepts randomization applets in which the randomness is specified using independent sliders (i.e., no random point, random element, etc., but also no dynamic boundaries based on other parameters). Although assigning values to parameters in the plugin is neither mandatory nor required, doing so will designate a fixed parameter value to the solver's task and the submitted solution will have these initial parameters assigned. If we

don't, the solutions will be regenerated with new parameter values each time they are opened, resulting in solutions that appear different each time. It is recommended to let Moodle manage the randomization parameters to avoid this.

3.1.1. Solution to problem number 1

What can we do if the randomization parameters cannot be made independent of each other? In the case of a triangle construction task when three sides are given, (https://www.geogebra.org/m/vd8qt88s), we cannot do so arbitrarily because, in order to satisfy the triangle inequality, the limits of the third side must depend on the values of the other two sides. When considering triangle inequality, we cannot have three random values for the lengths of the three sides, as this could result in a triangle that cannot be constructed. Two lengths of sides (a and b) can be adjusted to change in an interval between a given minimum and maximum value, while the third length (c) moves between the values $|a - b| + 1$ and $a + b - 1$. It is enough for the value to be between $|a - b|$ and $a + b$. However, it is important to note that this range can sometimes lead to the formation of a visually unappealing triangle, characterized by one large obtuse angle and two small acute angles, particularly when the third length approaches the minimum or maximum values.

Due to the non-independence of the third length (c), the website was unable to handle such a construction problem. Even though GeoGebra can process these independent and dependent variables, the GeoGebra applet (which was plugged into the system) did not function. In order to resolve this issue, a randomSeed variable was used in the form of a hidden slider (as shown in Figure 2). Since every problem can contain a Refresh button (as shown in Figure 3), this randomSeed variable is also beneficial for generating a new construction problem by refreshing the randomSeed variable when the Refresh button is used in the Moodle integrated applet. The Refresh button (as shown in Figure 4) must be implemented independently because the applet's Refresh button corrupts the randomseed, and in Moodle, only the randomseed parameter is provided.

The subsequent problematic area was the process of reopening a previously saved construction problem. Ideally, a construction problem that has been started, saved, closed, and reopened should contain the same values as before, without the need to generate new values by refreshing the randomSeed variable.

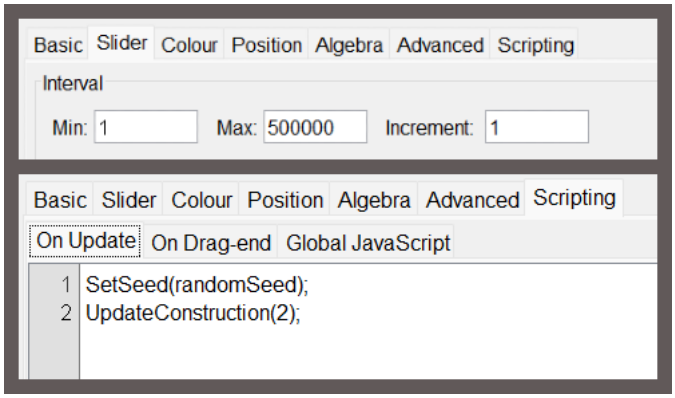

Figure 2. Randomseed slider (hidden) parameters and On update script.

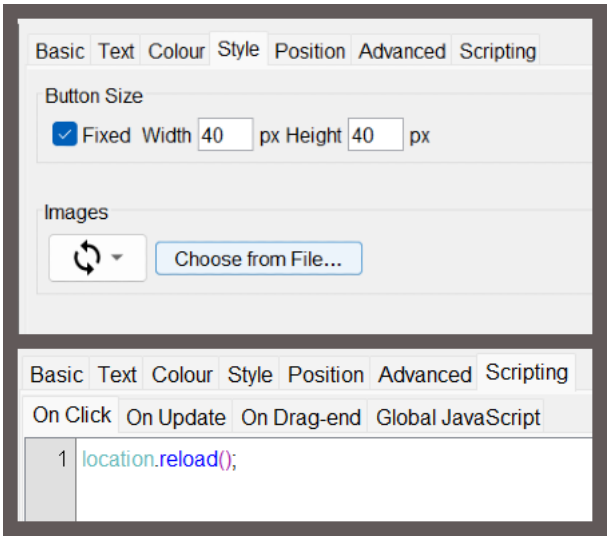

Figure 3. Refresh button with refresh icon (visible) and On click script.

```
function ggbOnInit() {
    ggbApplet.registerAddListener("newObjectListener");
    ggbApplet.evalCommand("RunUpdateScript(randomSeed)");
    ggbApplet.showResetIcon(false);
    ggbApplet.evalCommand("SetCoords(refreshButton, x(Corner(5))-50, 10)")
}

function newObjectListener(obj) {
    if (obj != "finished1" && obj != "finished2") {
        // checking polygons with the AreEqual command
        finished1 = ggbApplet.getValueString("finished1");
        if (finished1.indexOf("true") == -1) {
            var cmd = "finished1 = AreEqual(" + obj + ", answer1)";
            ggbApplet.evalCommand(cmd);
        }
        finished2 = ggbApplet.getValueString("finished2");
        if (finished2.indexOf("true") == -1) {
            var cmd = "finished2 = AreEqual(" + obj + ", answer2)";
            ggbApplet.evalCommand(cmd);
        }
        // display of the final result
        finished1 = ggbApplet.getValueString("finished1");
        finished2 = ggbApplet.getValueString("finished2");
        if (finished1.indexOf("true") > -1 || finished2.indexOf("true") > -1) {
            ggbApplet.setVisible("result", true);
        }
    }
}
```

Figure 4. Three extra lines added to ggbOnInit in Global JavaScript.

3.1.2. Solution to problem number 2

The second problem arose from having multiple correct solutions; for instance, if a parallelogram is to be constructed given one of its sides, the altitude of this side, and the length of the other side (https://www.geogebra.org/m/pyb2msth). In this case, we have to construct each correct solution and then consider these polygons (all four) as the target objects. In a modified script (as shown in Figure 5), we create logical variables for each target object. If the solver constructs one of the correct solutions, i.e., selects an identical polygon, the corresponding logical variable is initialized prior to the start of construction and is assigned a true logical value when the correct rectangle is constructed. We can already evaluate these logical variables using Moodle.

```
function ggbOnInit() {
    ggbApplet.registerAddListener("newObjectListener");
    ggbApplet.evalCommand("RunUpdateScript(randomSeed)");
    ggbApplet.showResetIcon(false);
    ggbApplet.evalCommand("SetCoords(refreshButton, x(Corner(5))-50, 10)")
}

function newObjectListener(obj) {
    if (obj != "finished1" && obj != "finished2" && obj != "finished3" && obj != "finished4") {
        // checking polygons with the AreEqual command
        finished1 = ggbApplet.getValueString("finished1");
        if (finished1.indexOf("true") == -1) {
            var cmd = "finished1 = AreEqual(" + obj + ", answer1)";
            ggbApplet.evalCommand(cmd);
        }
        finished2 = ggbApplet.getValueString("finished2");
        if (finished2.indexOf("true") == -1) {
            var cmd = "finished2 = AreEqual(" + obj + ",  answer2)";
            ggbApplet.evalCommand(cmd);
        }
        finished3 = ggbApplet.getValueString("finished3");
        if (finished2.indexOf("true") == -1) {
            var cmd = "finished3 = AreEqual(" + obj + ",  answer3)";
            ggbApplet.evalCommand(cmd);
        }
        finished4 = ggbApplet.getValueString("finished4");
        if (finished2.indexOf("true") == -1) {
            var cmd = "finished4 = AreEqual(" + obj + ",  answer4)";
            ggbApplet.evalCommand(cmd);
        }
        // display of the final result
        finished1 = ggbApplet.getValueString("finished1");
        finished2 = ggbApplet.getValueString("finished2");
        finished3 = ggbApplet.getValueString("finished3");
        finished4 = ggbApplet.getValueString("finished4");
        if (finished1.indexOf("true") > -1 || finished2.indexOf("true") > -1 || finished3.indexOf("true") > -1 || finished4.indexOf("true") > -1) {
            ggbApplet.setVisible("result", true);
        }
    }
}
```

Figure 5. Global JavaScript for automatic evaluation.

3.1.3. Solution to problem number 3

The solution to the third problem has already been mentioned, but we will elucidate it in greater detail now. When creating a task, we can adjust the size of the workspace, the placement and rotation of the objects we've added, and the limits of the sliders of the parameters we've specified so that the solver can fit all of the required construction parts and details into the workspace we've specified. There is no universal solution to this problem; we can only achieve it through a combination of the above so that we optimize the applet for the given task or even fine-tune it continuously by adjusting the above parameters so as not to detract from the user experience and make it impossible for the student to complete the assignment by not fitting into the workspace. In particular, for the second applet's other two tangent points, which were found by rotating the fixed point on the circle, we set the limits of the angles of rotation so that, for the maximum radius, the tangent taken at the fixed point is still in the workspace, as is the intersection of these two tangents. First, we guessed the finite values of the interval of angles, then we modified them by trial and error to meet our expectations.

In addition, the system awards the students points for solving a construction problem, and teachers can assign grades based on these points.

4. Conclusions

With the aid of our free web-based mathematics assessment tool, we can assess and measure the success rate of primary school students in solving geometric construction problems, an area currently lacking in national and international assessments. Furthermore, while automatic evaluation applets in GeoGebra are very useful on their own or in interactive mathematics textbooks, as they provide plenty of interactivity and feedback during the solving process, integrating such applets into Moddle LMS will greatly benefit mathematics teachers. By integrating geometric construction applets with automatic evaluation into an LMS such as Moodle, teachers can track students scores and progress, manage students, and compile exercise or homework exercises for them according to their needs in a much less time-consuming fashion. Moreover, as the advanced use of dynamic geometry systems is part of the standard curriculum for mathematics teacher education, the findings of the paper can be effectively demonstrated to prospective teachers. In addition, it may be interesting and useful to consider if there are multiple correct solutions to a problem, which one(s) of these will be constructed by the students, and whether the stereotype that solutions 'below the line' in the opposite half-plane will not be considered or will be less likely constructed by the students is true. Even though we now have such randomized geometric construction problems with automatic evaluation, it is still challenging to get a significant number of schools to deviate from their traditional methods, use these tools, and incorporate them at some point in their mathematics lessons.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was funded by VEGA grant (Vedecká Grantová Agentúra MŠVVaŠ SR) number 1/0386/21.

Conflict of interest

The authors declare no conflict of interest.

References

1. Euclid, *Euclid's Elements of Geometry*, 2008. ISBN 978-0-6151-7984-1.
2. G. Polya, *Mathematical Discovery*, John Wiley & Sons, 1981. ISBN 0-471-08975-3.
3. A. Levav, R. Leikin, Multiple Solutions for a Problem: A Tool for Evaluation of Mathematical Thinking in Geometry, *CERME 6*, (2010), 776–785.
4. M. Venturini, N. Sinclair, Designing Assessment Tasks in a Dynamic Geometry Environment, *MEDE*, **8** (2017). https://doi.org/10.1007/978-3-319-43423-0_5
5. *Mathematical Association of America: WebWork*, Available from: `https://openwebwork.org/`
6. *GeoGebra*, Available from: `https://www.geogebra.org/`
7. O. Birgin, F. Topuz, Effect of the GeoGebra Software-Supported Collaborative Learning Environment on Seventh Grade Students' Geometry Achievement, Retention and Attitudes, *J. Educ. Res.*, **114** (2021), 474–494. https://doi.org/10.1080/00220671.2021.1983505
8. H. Zulnaidi, E. Oktavika, Hidayat, Effect of use of GeoGebra on achievement of high school mathematics students, *Educ. Inf. Technol.*, **25** (2020), 51–72. https://doi.org/10.1007/s10639-019-09899-y
9. N. Arbain, A. N. Shukor, The Effects of GeoGebra on Students Achievement, *Procedia-Social Behav. Sci.*, **172** (2015), 208–214. https://doi.org/10.1016/j.sbspro.2015.01.356
10. Y. Zengin, H. Furkan, T. Kutluca, The Effect of Dynamic Mathematics Software GeoGebra on Student Achievement in Teaching of Trigonometry, *Procedia-Social Behav. Sci.*, **31** (2012), 183–187. https://doi.org/10.1016/j.sbspro.2011.12.038
11. K. A. Bakar, A. F. M. Ayub, R. Mahmud, Effects of GeoGebra towards Students' Mathematics Performance, *ICREM7*, (2015), 180–183. 10.1109/ICREM.2015.7357049
12. S. Radovic, M. Radojicic, K. Veljkovic, M. Maric, Examining the effects of Geogebra applets on mathematics learning using interactive mathematics textbook, *Interact. Learn. Environ.*, (2020), 32–49. 10.1080/10494820.2018.1512001
13. L. Frenken, P. Libbrecht, B. Becker, G. Greefrath, Dynamic Geometry Tasks in Standardized Assessment–Analysis of Solution Processes and Consequences for Practice, *IJMEST*, (2022). https://doi.org/10.1080/0020739X.2022.2036838
14. Š. Gergelitsová, T. Holan, GeoTest-A system for the automatic evaluation of geometry-based problems, *Comput. Appl. Eng. Educ.*, (2016), 297–304. 10.1002/cae.21712

15. Š. Gergelitsová, T. Holan, An Automatic Evaluation of Construction Geometry Assignments, *EC-TEL*, **7563** (2012). https://doi.org/10.1007/978-3-642-33263-0_41

16. T. P. Dick, Using Dynamic CAS and Geometry to Enhance Digital Assessments, *ICME-13 Monogr.*, (2018), 267–287. https://doi.org/10.1007/978-3-319-76575-4_14

17. *Moodle*, Available from: `https://moodle.org/`

Supplementary

Script from Figure 1

```
function ggbOnInit() {
    ggbApplet.debug("ggbOnInit");
    ggbApplet.registerAddListener("newObjectListener");
}

function newObjectListener(obj) {
    if (obj != "finished") {
        var cmd = "finished = (" + obj + "==target)";
        ggbApplet.debug(cmd);
        ggbApplet.evalCommand(cmd);
        finished = ggbApplet.getValueString("finished");
        if (finished.indexOf("true") > -1) {
            ggbApplet.setVisible("solution", true);
        }
    }

}
```

Script from Figure 4

```
function ggbOnInit() {
    ggbApplet.registerAddListener("newObjectListener");
    ggbApplet.evalCommand("RunUpdateScript(randomSeed)");
    ggbApplet.showResetIcon(false);
    ggbApplet.evalCommand("SetCoords(refreshButton, x(Corner(5))-50, 10)")
}

function newObjectListener(obj) {
    if (obj != "finished1" && obj != "finished2") {
        // checking polygons with the AreEqual command
        finished1 = ggbApplet.getValueString("finished1");
        if (finished1.indexOf("true") == -1) {
            var cmd = "finished1 = AreEqual(" + obj + ", answer1)";
            ggbApplet.evalCommand(cmd);
```

```
        }
        finished2 = ggbApplet.getValueString("finished2");
        if (finished2.indexOf("true") == -1) {
            var cmd = "finished2 = AreEqual(" + obj + ", answer2)";
            ggbApplet.evalCommand(cmd);
        }
        // display of the final result
        finished1 = ggbApplet.getValueString("finished1");
        finished2 = ggbApplet.getValueString("finished2");
        if (finished1.indexOf("true") > -1 || finished2.indexOf("true") > -1) {
            ggbApplet.setVisible("result", true);
        }
    }
}
```

Script from Figure 5

```
function ggbOnInit() {
    ggbApplet.registerAddListener("newObjectListener");
    ggbApplet.evalCommand("RunUpdateScript(randomSeed)");
    ggbApplet.showResetIcon(false);
    ggbApplet.evalCommand("SetCoords(refreshButton, x(Corner(5))-50, 10)")
}

function newObjectListener(obj) {
    if (obj != "finished1" && obj != "finished2" && obj != "finished3"
     && obj != "finished4") {
        // checking polygons with the AreEqual command
        finished1 = ggbApplet.getValueString("finished1");
        if (finished1.indexOf("true") == -1) {
            var cmd = "finished1 = AreEqual(" + obj + ", answer1)";
            ggbApplet.evalCommand(cmd);
        }
        finished2 = ggbApplet.getValueString("finished2");
        if (finished2.indexOf("true") == -1) {
            var cmd = "finished2 = AreEqual(" + obj + ",  answer2)";
            ggbApplet.evalCommand(cmd);
        }
        finished3 = ggbApplet.getValueString("finished3");
        if (finished2.indexOf("true") == -1) {
            var cmd = "finished3 = AreEqual(" + obj + ",  answer3)";
            ggbApplet.evalCommand(cmd);
        }
        finished4 = ggbApplet.getValueString("finished4");
        if (finished2.indexOf("true") == -1) {
```

```
            var cmd = "finished4 = AreEqual(" + obj + ",  answer4)";
            ggbApplet.evalCommand(cmd);
        }
        // display of the final result
        finished1 = ggbApplet.getValueString("finished1");
        finished2 = ggbApplet.getValueString("finished2");
        finished3 = ggbApplet.getValueString("finished3");
        finished4 = ggbApplet.getValueString("finished4");
        if (finished1.indexOf("true") > -1 || finished2.indexOf("true") > -1 ||
        finished3.indexOf("true") > -1 || finished4.indexOf("true") > -1) {
            ggbApplet.setVisible("result", true);
        }
    }
}
```

AIMS Press