Tim Lutz, Guido Pinkernell and Chris Sangwin
In: Contributions to the International Meeting of the STACK Community 2023. TTK University of Applied Sciences: Tallinn, Estonia.

GeoGebra integrated in STACK

Tim Lutz[1], Guido Pinkernell, Chris Sangwin

Abstract: The first international STACK conference took place in 2018 at the University of Erlangen-Nuremberg in Fürth, Germany. There, one speaker presented a method developed for the conference to create graphically interactive randomized STACK tasks using GeoGebra, called GeoGebraSTACK_HelperTool (Lutz 2019). The new integration called "GeoGebra in STACK" makes this feature available directly within STACK core, so there is now a GeoGebra based way to create graphically demanding tasks, with the possibility of giving adaptive feedback within STACK. In addition, all applets from the GeoGebra material library are available for the new application. For more advanced task creators, "GeoGebra in STACK" provides documented commands to gain more control of the communication between GeoGebra and STACK. Link to GitHub repository and documentation: https://tim-lutz.de/STACK23

Keywords: GeoGebra, GeoGebra in STACK

1 Application scenarios of the combination of GeoGebra and STACK

At the first international STACK conference Tim Lutz presented the self-developed GeoGebraSTACK_HelperTool. The software, written in the Java language, allows linking GeoGebra and STACK with the help of a simply designed graphical user interface. Lutz (2019) shows a number of example scenarios for which the combination of GeoGebra and STACK can be used:

- GeoGebra graphical applets for displaying geometric information:
  showing, not listening to or setting objects via STACK
- GeoGebra graphical applets for randomized graphics:
  showing and manipulating applets via STACK
- GeoGebra graphical applets for interactive tasks:
  listening to objects in applets to calculate feedback based on values of these objects

In Lutz (2019), on the other hand, the presentation of mathematical content components to describe useful combinations of GeoGebra and STACK was not made explicit.

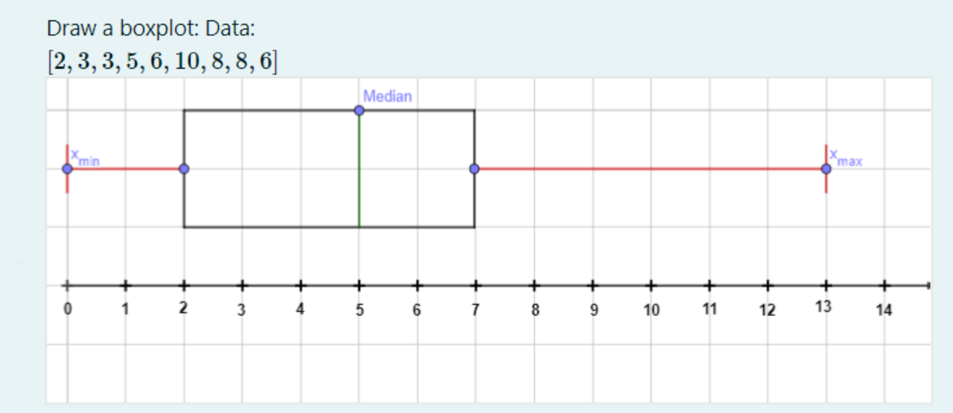
In the following, we will make explicit a content area that does not come from geometry or work with functions, but where GeoGebra can still be used mathematically to create

[1] an Erasmus-project of University of Education Heidelberg, Germany in cooperation with University of Edinburgh.

DOI 10.5281/zenodo.8133499

interactive tasks: data and randomness

GeoGebra can be used to create many types of graphs. In STACK it can be used to read and (partially) create diagrams (Fig. 1).



Draw a boxplot: Data:

$[2, 3, 3, 5, 6, 10, 8, 8, 6]$

Fig. 1: Example task: "Draw" a boxplot by dragging some points.

All examples shown are provided as STACK questions for Moodle via tim-lutz.de/STACK23

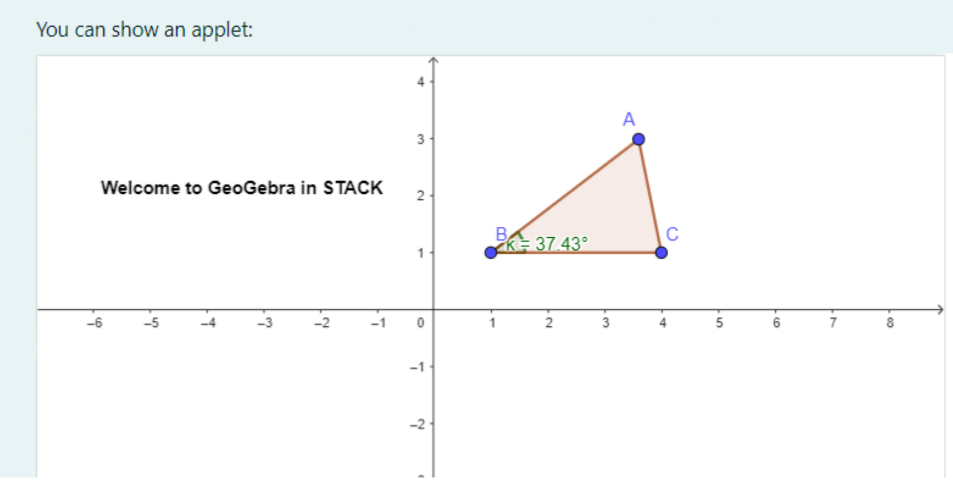
In STACK, the data series are randomized. The question is answered by interaction/manipulation of the GeoGebra applet. Feedback is determined based on the values passed from the applet to STACK in the form of the location of the shifted points. Similarly, tasks are easy to implement, such as drawing and reading bar graphs. A collection of tasks in the style of the example task with a focus on primary education is currently being developed by students as part of their master's thesis at the RPTU in Landau, Germany. Therefore the developed tasks demonstrate that GeoGebra applets can also be useful to create more general graphical user interfaces for STACK tasks. After completion, these tasks will be made available in German and English via tim-lutz.de/STACK23

2 Advantages and design principles of the block-based approach

"GeoGebra in STACK" uses the block design of STACK, as it was already possible for JSXGraph. With "GeoGebra in STACK", tasks can be created without the knowledge of the JavaScript programming language, as was already the case with the GeoGebraSTACK_HelperTool. One advantage of this technically new approach as a block for integrating GeoGebra into STACK is that tasks can thus be edited directly on the STACK user interface. Since STACK tasks are usually revised many times due to their complexity, this eliminates the need to go back and forth between GeoGebra, the helper tool and STACK, in contrast to using the GeoGebraSTACK_HelperTool. "GeoGebra in STACK" prefers to edit GeoGebra files directly on geogebra.org. If the GeoGebra file changes there, the change is directly visible in the STACK task, an additional mediator in the form of the HelperTool is therefore no longer necessary with online Moodle servers. The GeoGebra block code is written directly into the question text. The HTML editor mode should always be used. It should be noted that naming conventions for variables must be followed: points must be written in uppercase and values in lowercase. A list of rules for naming objects can be found in the documentation: https://tim-lutz.de/STACK23

Examples of "GeoGebra in STACK" "block"-code

```
<p>You can show an applet:</p>
[[geogebra]]
params["material_id"]="seehz3km";
[[/geogebra]]
```



You can show an applet:

Fig. 2: technical example: generated by the code above.

This code shows below the text "You can show an applet" the GeoGebra file with material_id "seehz3km"(Fig. 2). In the applet seehz3km the points A and B exist. If you want to move the coordinates of these points to another place via STACK while loading the task, you define question variables in STACK:

```
A:[2,3]; B:[1,2];
```

In addition, you tell GeoGebra to fill the points A and B with the STACK variables A and B by adding the "set" subtag to the code above:

```
[[geogebra set="A,B"]]
params["material_id"]="seehz3km";
[[/geogebra]]
Write the coordinates of \(A\):
   [[input:ans1]][[validation:ans1]]
```

If instead you want to observe where the task handler placed the free-moving point A, and what the value b is, you would instead add the "watch" subtag as follows

```
[[geogebra watch="A,b"]]
params["material_id"]="seehz3km";
[[/geogebra]]
[[input:A]][[validation:A]]
[[input:b]][[validation:b]]
```

Also, the input code has been replaced. This is because GeoGebra variables that you want to watch with the watch subtag to be able to use them in STACK fulfill the same property as an algebraic expression entered by the task handler. A and B are restored on loading, so even after the task is completed or the input is saved, the task submission can be reconstructed.

To use the task, float values must be allowed for A and b in the input settings. More instructions, complete examples and the explanation of the remember subtag for efficient recovery of app states can be found in the documentation.

2.1 Comparison with the JSXGraph Implementation and the HelperTool

"GeoGebra in STACK" is based on the block-based code of the JSXGraph implementation for STACK. All JavaScript functions provided by JSXGraph are also available for "GeoGebra in STACK", especially the commands "stack_geogebra_bind_point(args)" etc.

In contrast to the JSXGraph implementation, a lot less JavaScript code needs to be looked at and adapted for many task ideas created with "GeoGebra in STACK". If one only sets values or wants to read them from GeoGebra, the required code always has a length of 3 lines, in which one only has to exchange the material_id for another one, complemented by the input names of the variables to be read from GeoGebra. This is possible, because by clever choice of naming conventions, "GeoGebra in STACK" interprets settings automatically, which had to be chosen manually in the JSXGraph implementation. This makes the STACK part of "GeoGebra in STACK" much easier to use than the corresponding JSXGraph code. Also, and arguably more importantly, there are many more people in the teacher education field who already use GeoGebra and can customize applets with GeoGebra, which is in line with the concern of the GeoGebraSTACK_HelperTool. In particular, the creation of GeoGebra in STACK tasks with student teachers only becomes feasible with reasonable effort because of the integration with STACK and the use of GeoGebra, as trials in material development seminars at RPTU, Landau have shown. With the material collection on geogebra.org, there is also a very wide variety of GeoGebra materials already created by other users that are freely available for non-commercial purposes.

In contrast to GeoGebraSTACK_HelperTool, no additional software needs to be installed, the whole task creation process is done in STACK or GeoGebra as web applications.

Advantage of JSXGraph is that the necessary files are very small compared to GeoGebra, i.e. in case of poor internet connection and low-powered device JSXGraph applications load faster. JSXGraph may be used commercially without further restrictions. Moreover, JSXGraph elements are independent of other servers, while the preferred workflow for creating tasks in "GeoGebra in STACK" described below assumes the availability of materials on geogebra.org.

3 Describing a workflow for creating "GeoGebra in STACK" tasks

The following two tasks were developed (and translated for this article) as part of a seminar on creating interactive graphically randomized tasks with university students at the primary level. They are suitable to explain the workflow of creating tasks with GeoGebra in STACK.

Origin: In primary education, learning to link digitally and analogously represented clock times is an educational task.

Task idea 1: An analog clock is to be displayed. The student has to write a digital time in prepared input fields.

1. Research on geogebra.org

First, a search on geogebra.org takes place with the goal of finding possibly already existing materials. Quickly a multirepresentation system is found by the user "Duane Habecker".

2. Copy the applet and adapt it to the needs of the STACK task

Once an applet is found, it usually needs to be adapted a bit, for example objects need to be hidden or renamed. For this purpose the applet should be copied and saved to materials. The result (Fig. 3) is for example the applet https://www.geogebra.org/m/q9tv7crg

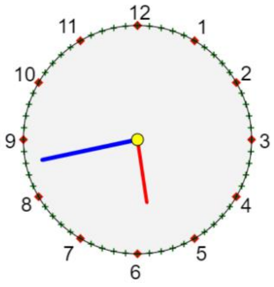


Fig. 3: Screenshot of modified applet: here is nothing to interact and many objects were hidden.

3. Analysis of the applet for STACK

Furthermore, an analysis of the applet has to be done: Which elements should be controlled by STACK, which should be observed. In the example we notice that with the GeoGebra variable "minutespassed" the analog time can be controlled. Therefore we define this variable with the same name in STACK and randomize it. We also add set="minutespassed" to the GeoGebra in STACK code.

4. Finishing the STACK task part

The hours and minutes can then be calculated from minutespassed in STACK and entered as a solution for input fields and feedback trees to be prepared in STACK.

Task idea 2: A digital time is given, the student has to set the clock hands by grabbing and moving the hands.

1. Research on geogebra.org

The same multi-representation system of "Duane Habecker" is used in this example.

2. Copy the applet and adapt it to the needs of the STACK task

The modified applet (Fig. 4) can be found here: https://www.geogebra.org/m/g87btwsb

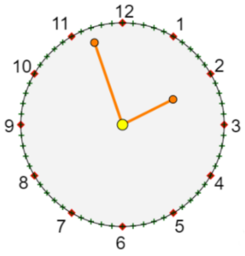


Fig. 4: Screenshot of modified applet: here you can grab the points to change the analog display:

3. Analysis of the applet for STACK

In the GeoGebra variables k and l angle elements were added in GeoGebra, which can be read out as radians values. Therefore, the task creator adds the inputs k and l in STACK to be able to process the angles of the clock hands in STACK. Be sure to allow "floats".

4. Finishing the STACK task part

In STACK, hours and minutes are now randomized in a different way than in the implementation of task idea 1. Two angles can then be calculated from hours and minutes and compared with the angles of the student answer that are stored in k and l. The angles of the clock hands can then be processed in STACK. After adding l and k in GeoGebra and reloading the STACK task page, the modified applet can be found there, a big advantage of using "GeoGebra in STACK" compared to the GeoGebraSTACK_HelperTool. At this point one has to decide whether to allow a deviation in answer checking in STACK or whether to define the objects in GeoGebra as "fixed to grid" instead, depending on the task.

At the latest now the task creator becomes aware that there are several possibilities how this task can be solved. Is the hour hand at 2:30 p.m. between 2 and 3 o'clock, or would one expect it to be at 2 o'clock. At these and other occasions, the task creator is thus confronted with mathematical didactic reflection that he might not have noticed if he had only looked at corresponding tasks in a textbook. This specific consideration is meant to indicate the potential that the creation of digital tasks has for the analysis of analog task types, because reasoned decisions have to be made all the time, be it in the feedback tree, be it in the definition of the consequences of an interaction with the applet.

4 Summary and Outlook

As part of the Erasmus project AuthOMath, Tim Lutz’s GeoGebraSTACK_HelperTool has now been completely revised and integrated into the core of STACK. Under the name "GeoGebra in STACK" the tool is now available as a new block element, in which the communication between GeoGebra and STACK is dynamically automated.

As seen from the secondary and primary examples, the rich selection of applets from GeoGebra.org alone results in many high-quality graphical designs that only need to be modified slightly, if at all, for use in STACK. "GeoGebra in STACK" makes it easier than ever to incorporate such elements. So easy, in fact, that prospective teachers are already using "GeoGebra in STACK" in material creation seminars.

When implementing task ideas, a guiding principle should always be followed, anything graphical or geometric should be implemented in GeoGebra, hidden there if necessary, and then processed algebraically by STACK via setting points and values and observing points or values to make feedback decisions.

In the form of a regular STACK update, the features of "GeoGebra in STACK" will be made available to all STACK users during 2023 and will be usable without additional installations in the core of STACK.

In order to make the use of "GeoGebra in STACK" even more intuitive for beginners, a GUI will be added to STACK after the initial release, which will allow the GeoGebra block-code to be generated with graphical support as well.

5 Acknowledgements

The project is co-funded by the European Union under the Erasmus+ Programme, No. 2021-1-DE01-KA220-HED-000032031. This publication reflects only the authors' view. The Commission is not responsible for any use that may be made of this publication.

6 References

Lutz, T. (2019). GeoGebra and STACK. Creating tasks with randomized interactive objects with the GeoGebraSTACK_HelperTool. https://doi.org/10.5281/zenodo.3369599