designs

CITESCORE 3.9

Article

Contribution of Artificial Intelligence (AI) to Code-Based 3D Modeling Tasks

Marianna Zichar and Ildikó Papp

Special Issue
Design Process for Additive Manufacturing

Edited by
Dr. Paweł Turek

MDPI

https://doi.org/10.3390/designs8050104

designs

Article

Contribution of Artificial Intelligence (AI) to Code-Based 3D Modeling Tasks

Marianna Zichar * and **Ildikó Papp**

Department of Data Science and Visualization, Faculty of Informatics, University of Debrecen, H-4028 Debrecen, Hungary; papp.ildiko@inf.unideb.hu
* Correspondence: zichar.marianna@inf.unideb.hu

Abstract: The rapid advancement of technology and innovation is also impacting education across different levels. The rise of Artificial Intelligence (AI) is beginning to transform education in various areas, from course materials to assessment systems. This requires educators to reconsider how they evaluate students' knowledge. It is crucial to understand if and to what extent assignments can be completed using AI tools. This study explores two hypotheses about the risks of using code-based 3D modeling software in education and the potential for students to delegate their work to AI when completing assignments. We selected two tasks that students were able to successfully complete independently and provided the same amount of information (both textual and image) to AI in order to generate the necessary code. We tested the widely used ChatGPT and Gemini AI bots to assess their current performance in generating code based on text prompts or image-based information for the two models. Our findings indicate that students are not yet able to entirely delegate their work to these AI tools.

Keywords: 3D modeling; 3D printing; education; OpenSCAD; artificial intelligence; computational thinking; creativity; ChatGPT; Gemini

check for updates

Citation: Zichar, M.; Papp, I. Contribution of Artificial Intelligence (AI) to Code-Based 3D Modeling Tasks. *Designs* **2024**, *8*, 104. https://doi.org/10.3390/designs8050104

Academic Editor: Paweł Turek

Received: 8 August 2024
Revised: 14 October 2024
Accepted: 16 October 2024
Published: 18 October 2024

1. Introduction

The technological development of Industry 4.0 has dramatically impacted individual industries, resulting in a significant increase in productivity. Technological adaptation has become a critical factor because missing out on new technological achievements can cause a competitive disadvantage. The process is characterized by a high degree of digitization. It can be associated with many concepts known today, such as Additive Manufacturing, Artificial intelligence, Augmented reality, Autonomous robots, Big data and analytics, Blockchain, Cybersecurity, IoT (Internet of Things), and Simulation [1]. The shift towards Industry 5.0 can be considered a paradigm shift since instead of focusing on the role of technology, it places the cooperation between technology and people at the center of development. In addition, Industry 5.0 is closely linked to efforts for sustainable and inclusive economic growth [2].

Educational systems must reflect technical changes since they aim to bring up future employees (engineers, IT professionals) for the professional challenges ahead. For this reason, many institutions are motivated to teach 3D printing to students. However, creating a 3D model is necessary for 3D printing, so 3D modeling and printing often go hand in hand. Artificial intelligence, one of the latest emerging technologies, already provides significant assistance in various areas, such as writing texts, generating images, and even coding. If teachers want to assess the students' skills rather than those of the AI tools, then it is essential to be aware of the current capabilities of these systems. This study evaluates the abilities of two of the most popular AI bots in creating 3D models in OpenSCAD.

1.1. Related Work

Experience in 3D printing technologies helps to implement it in many aspects of their daily education process. The beginning can be a geometric modeling course, followed later by a CAD course to develop 3D models of the prototypes. This course can be suitable for introducing various technologies, applications, and practical experience related to additive manufacturing [3].

The best method is the learning-by-doing method when the motivation for acquiring technical knowledge is provided by processing a topic or solving a project task [4]. The goal may not only be that the students learn about and be able to apply modern technologies but also that an ancient and traditional subject, mathematics, be taught in a more modern form and be brought closer to the world of today's students [5]. Implementing STEAM (Science, Technology, Engineering, Arts, and Mathematics) lessons into regular university courses is difficult because of scheduled classes with time limits during the semester. To break these barriers, there are a lot of different suggested pedagogic instructions, methods, and case studies [6,7]. At the same time, it can already be seen that the targeted development of certain skills—for example, computational thinking—can be helpful even at a younger age [8]. It has been proven in many areas that project tasks carried out in groups would also have an effect on reducing social differences and help create an inclusive atmosphere [9].

Various excellent projects involve collaborations between scientists, engineers, and artists to provide innovative solutions that impact society. In accordance with the most essential guidelines of modern museology, the expectations connected to the museum have also changed. In the museum of the 21st century, an interactive exhibition is needed, in which digitization and some kind of creative activity for visitors play an increasingly important role [10].

Technological knowledge cannot be used for its own sake; it is important to use it to meet unique needs, so that it has a more significant impact on society [9,11].

Companies, educational institutions, and other associations can establish so-called Makerspaces, where people can use different tools to make their projects. The sessions help to reduce stress and increase creativity while introducing techniques that may not have been known before [12,13].

The critical aspects of 3D modeling for the authors are related to solid modeling, i.e., when a production process, typically 3D printing, can be started based on the model. Nowadays, it is typical for the model or a part of it to be made based on some 2D information. 3D solid modeling applications have some tools to transform 2D content into 3D objects. The 2D geometry content here comes from a drawing or a movement path. There are also online converters that are not connected to editing software, which can generate a 3D model in stl or obj format directly based on an image file [14]. In this case, the image's resolution greatly influences the output form's shape, accuracy, and resolution. Specialized model generation provided by artificial intelligence (AI) can be said to be novel; several services are available that can be used to create an interactively displayed product asset based on an image [15]. This way of getting the model is much cheaper and faster for the customer than using a professional 3D modeling service. We must mention that the model expectations of 3D production do not match the expectations necessary for visualization. BOOM Interactive's Bubbles application was developed for multiple operating systems to be widely used for the quick and accurate generation and furnishing of architectural interior design spaces [16]. This AI-assisted 3D space design tool is not only to create scenes from 2D floor plan images, scan their space, or build from scratch, but the real-time 3D editor allows them to make changes, such as moving walls, adding windows and doors, decorating, and making decisions. Meshy AI toolkit was developed to transform text or images into 3D models or texture to support artists, game developers, and creators with display-focused needs [17]. Some further online 3D model generators are overviewed in [18], but they aim to general 3D model that are not specific to 3D modeling and the file formats are also special ones which should be transform to stl with some other software.

1.2. Optional Course at Univesity of Debrecen

The Faculty of Informatics at the University of Debrecen offers students the opportunity to learn about 3D printing and modeling in the frame of an optional course. The fact that the Faculty of Informatics won a MakerBot Replicator 5th Generation 3D printer in a competition in 2016 gave an excellent motivation for working out the course syllabus [19]. Later, a Prusa MK2.5 printer and a Sense 3D scanner enriched the available tools during the classes. Both printers are of the Fused Deposition Modeling (FDM) type, which is the most popular printer type among hobby users. However, there are also many great examples of professional use of FDM devices in the industry. The main goal of the course is to introduce the 3D printing process, including designing a model, slicing it, and 3D printing. At the beginning of the course, we discuss the basics of FDM printing; the students get to know the printing process, the different materials that can be used for printing, and learn about the requirements for 3D printability. We can use various file repositories to obtain a model for 3D printing quickly and easily, but the course aims for students to create solid objects of varying complexity.

For our modeling purpose, we preferred applications that are available for free, so that everyone can participate in completing the course with equal chances.

1.2.1. TinkerCAD

In the first half of the course, we use the TinkerCAD application running in a web browser, which makes a large collection of primitives and improved primary bodies available after registration [20]. The complex, unique designs can be created by computing the union or difference of bodies, which means that limited CSG (Constructive Solid Geometry) modeling is applied (Figure 1). All private designs are stored in the personal account, and all the calculations run on servers of the owner Autodesk Inc. (San Francisco, CA, USA). The models can be downloaded for manufacturing, and projects can be shared and embedded in web pages. The TinkerCAD application has its shortcomings in terms of specifying all exact dimensions, the possibilities of redesigning models and engineering techniques, but it is nevertheless a perfect choice for learning entry-level modeling tools.

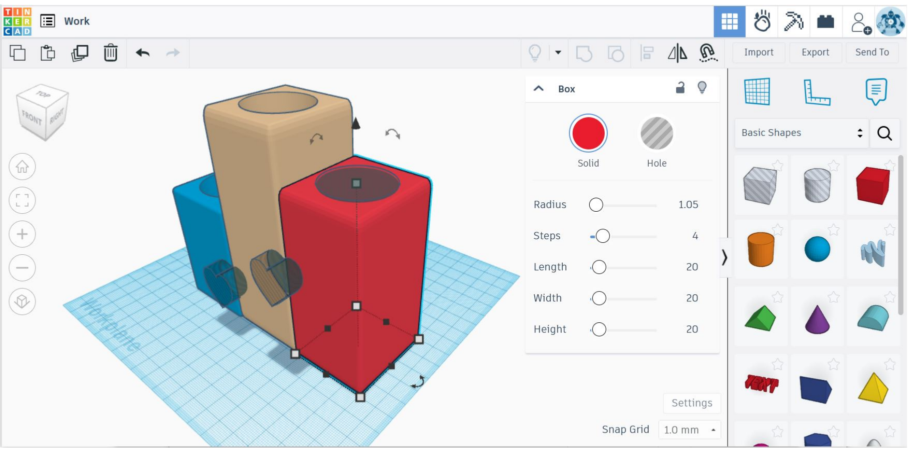


Figure 1. User interface of TinkerCAD application. Complex shapes can be designed from primitives of predefined libraries. All shapes can be used as solids or holes. Grouping is the tool to apply union or difference of the selected objects.

1.2.2. SolidWorks

The other modeling application chosen for the course is the SolidWorks engineering application, whose educational version is installed in our computer labs. At the same time, the Student Design Kit is available for students to download free of charge after completing a registration form [21]. SolidWorks is a CAD design system for mechanical engineering with many tools to help the user. Solids are created mainly from 2D drawings using various generation tools such as drawing, rotation, sweeping, or loft (Figure 2). Later, the basic shapes can be modified by cutting, chamfering, and filleting edges [22]. The modeling process is well documented, and each step can be reworked and modified afterward. As an engineering application, dimensioning is possible with a high degree of precision and parameterization, allowing model families to be created.

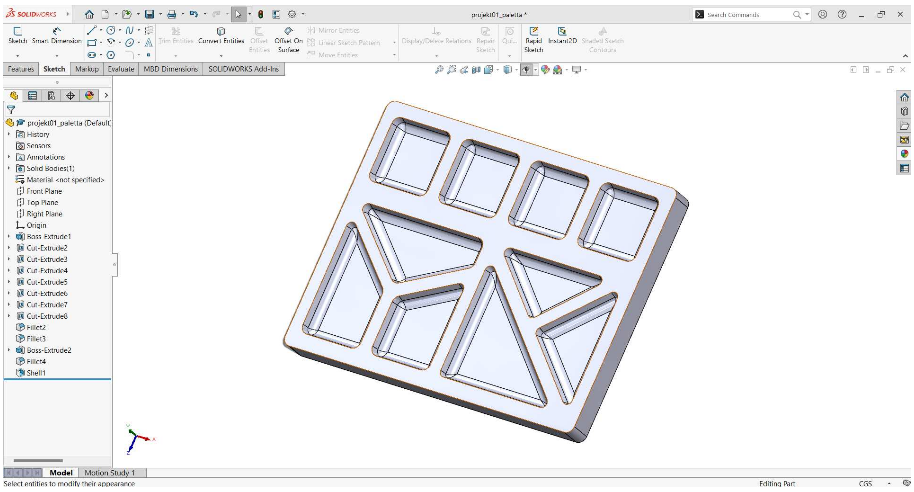


Figure 2. User interface of SolidWorks. The Command Manager Toolbar (horizontal bar at the top of the window) presents all available 3D Feature tools. The separated panel on the left shows tools building the model in the so-called design tree.

1.2.3. Print-in-Place—The Holy Grail of 3D Printing

Modeling is taught through targeted examples and exercises, and concepts specific to 3D printing are introduced in parallel. In preparation for 3D printing, students learn about the basic functionality of the slicing software and the different settings that affect print quality.

3D printing has unquestionable advantages over other manufacturing technologies. These include print-in-place object manufacturing, which is based on a model consisting of several precisely positioned parts connected with hinges and joints. The model is printed in a single printing process, still it will have moving parts without the need for any assembly (see Figure 3). The final model can contain even spinning parts.

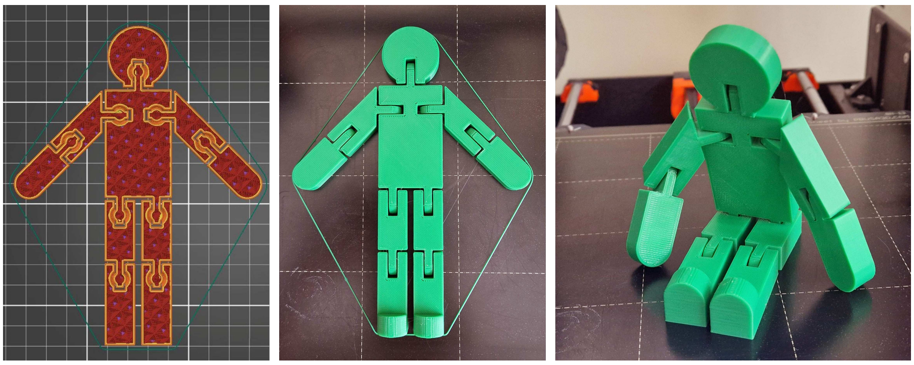


Figure 3. Sample print-in-place design. On the left is the model in the slicer software with removed top layers to illustrate separate parts with required gaps. The middle figure shows the printed object. The right figure shows the movement capability of this model immediately after the printing.

1.2.4. Evaluation of Students' Work

In the course, the evaluation of the students' work is based on three pillars: testing his theoretical knowledge, checking his modeling skills (one task per modeling software), and completing a project as a final assignment. The test on theoretical knowledge is realized in the E-learning (Moodle) system and focuses on basic facts, concepts, and terms and phrases of 3D printing and modeling. Basic skills gained using both 3D modelers are assessed in the modeling tests. Students must design models based on visual information (photos, images), instructions, and given dimensions within a specified time. They must upload the requested work files or shared links to the model, screenshots, and text supplements. The tests are organized in an exam-like environment, in a computer lab, with no supporting resources. By the end of the semester, students must complete their final assignment, which is the design of a 3D (FDM) printable object with detailed documentation. They can use all course materials and online content to demonstrate their proficiency and knowledge of the subject.

2. Research Hypotheses

Students who are offered the optional course mentioned above can program, so it is logical to consider whether code-based 3D modeling software should be included in their education or not. Fortunately, some free 3D modeling software, usually using their own scripting language, is available, thus enriching the curriculum is feasible.

2.1. OpenSCAD

3D modeling is most often done on a visual design interface, mainly with mouse manipulations and data entry, and the 3D model components can also be transformed and moved in the virtual space with mouse movements. OpenSCAD [23] takes a radically different approach to solid modeling. When we describe a component in some sentences, we need to answer some questions like what kind and what size of geometric elements it is made of, how they meet each other (whether they are in contact with each other or if there are gaps between them), are the components repeated or not, and if so, how many times and in what way, etc. The answers help to algorithmize the design process. OpenSCAD is a free, code-based, non-interactive modeler that does not focus on the artistic aspects of 3D modeling but instead on the CAD aspects. OpenSCAD uses its script language with control statements, named values (variables), modules, and functions (Figure 4). The software processes the code and generates a preview or rendered model version. It works with CSG modeling with primitives, 3D objects derived from 2D objects, and imported shapes.

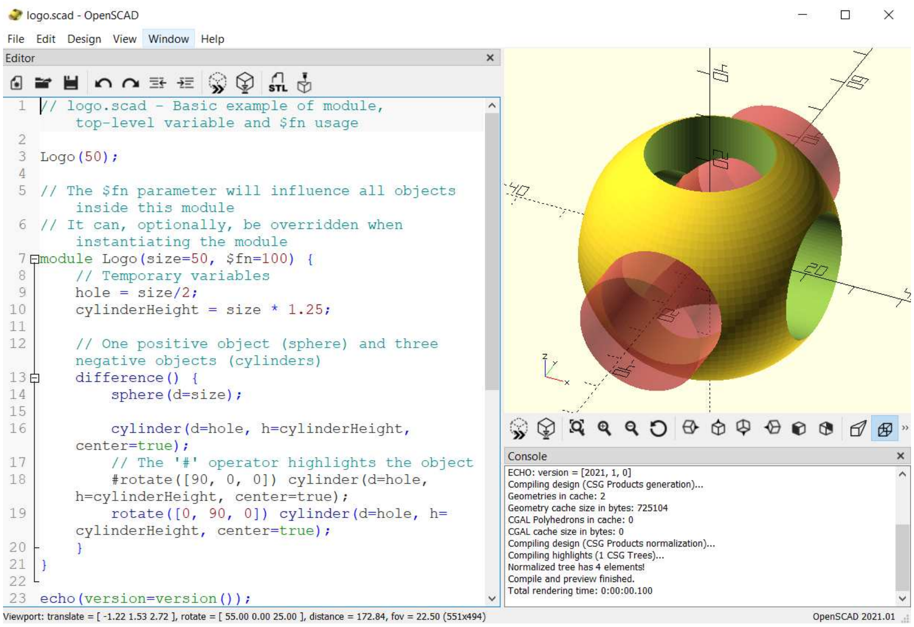


Figure 4. User interface of OpenSCAD application. Based on the sample code in the left window, the OpenSCAD logo is rendered.

2.2. Hypotheses

Lecturers must check students' knowledge, which can also be done as an assignment. In the third decade of the 21st century, when artificial intelligence chatbots are available with ever-expanding capabilities, we can wonder who will perform the tasks in the case of code-based 3D modeling software. Should we be afraid that the codes are not written by students but are generated by artificial intelligence? Can artificial intelligence help or hinder the work of students correcting partial solutions containing errors and misunderstood information?

In this study, we examine the following research hypotheses:

1. Introducing code-based 3D modeling software (such as OpenSCAD v.2021.01) for the students has the risk that they can generate the codes using only AI chatbots.
2. Currently available AI tools can provide relevant help in designing 3D models using a scripting language.

In the following, we will refer to these two hypotheses as H1 and H2, respectively.

3. Materials and Methods

In the frame of an ERASMUS+ project called INNOSID, we organized a workshop for students who were majored in IT-related programs at different universities in Croatia, France, Hungary, Portugal, and Spain [24]. The objective of our 5-h-long workshop was to introduce students with no previous knowledge of 3D printing and modeling to writing OpenSCAD scripts to form 3D printable models. Since the workshop took place in Valencia, Spain, we decided that the characteristics of the city would inspire the models. The idea promised multiple advantages:

- direct the students' attention to the architectural values of the city,
- develop their geometric skills,
- develop their algorithmic skills,
- motivate them to look for these shapes in the city after the workshop.

At the beginning of the workshop, the main principles and properties of 3D printing and the basics of OpenSCAD scripting (syntax rules, the most critical functions, flow control statements, and the role of modules) were discussed, followed by writing simple sample codes together. In the next stage, students received tasks to complete on their own. Each task consisted of a photo taken in the city of the object to be modeled and instructions about the primitives and functions to use. The 3D-printed models were also available in the classroom. At that time (May 2022), either ChatGPT [25], or another AI chatbot was not released; thus, the students worked alone.

To verify or drop the research hypotheses, we will check how two AI chatbots (ChatGPT from OpenAI LP (San Francisco, CA, USA) and Gemini [26] from Google AI (San Francisco, CA, USA)) can deal with two tasks of the students. There are other models, such as Grok-2, LLama-3.1, and Claude Sonnet, that we could test, but their popularity lags far behind the two leading ones. ChatGPT is considered the most well-known AI tool and is very popular; thus, its testing is unavoidable. Google, as a company with so many free services, is also very popular among students. The majority of the students have a Gmail account and use several Google tools. This was the reason to involve it in the testing. Since both companies make the temporary usage of their advanced AI models available, we used both the free versions and the advanced models. Prompts were given in English since the language of instruction in the INNOSID workshop was English. IT students are used to using English in their study and work, and we also run our courses at the university in English.

3.1. ChatGPT from OpenAI

ChatGPT is a chatbot based on the Generative Pre-trained Transformer (GPT) language model that was taught using texts from the Internet, Wikipedia, and digitized books. The system is based on deep learning technology and can communicate with humans in real time by allowing natural language processing and generating text responses to users' questions or requests. It mimics human-to-human communication at a high level, providing realistic and intelligent answers to user questions (prompts). The first version of ChatGPT was released in late 2022 based on the data collection until then. GPT-4 model has made ChatGPT more accurate, consistent, human-like, and detailed. The latest GPT-4o model, which can interpret images and browse the Internet, is available free of charge for a limited time only without a subscription.

3.2. Gemini from Google

Gemini is Google's AI-powered assistant, built right into Gmail, Docs, Sheets, and more. It offers enterprise-grade security and privacy. ChatGPT is the best solution for developer content and complex content requests. Gemini is a better solution for many images and creative content requests, as well as for built-in quality assurance tools.

4. Experiments

This section examines the workshop's two selected modeling tasks from the viewpoint of AI's applicability. Students solved these tasks, which proves that, based on the slides, the modeling tasks can be completed after a brief introduction to 3D modeling with OpenSCAD. We will also provide the slides containing the tasks and additional information that aided the students during the planning process. The two selected tasks have different characteristics. The first task involves a simple model geometry, but various modeling approaches can be applied. The second task has a much more complex model geometry, making it challenging to decipher the required solids and operators. To assist students with weaker geometry skills, the slides include a detailed explanation of the necessary steps to avoid failure.

Additionally, we will share our experience with the students' performance and our attempts to solve the tasks using artificial intelligence. We tested both the free and enhanced (image interpretation) versions. The ultimate goal was to generate the OpenSCAD code for both models.

4.1. Bridge Railing of Puente del Ángel Custodio

Although no river crosses the city of Valencia, it still has 18 bridges full of history over the Turia Garden, which was established in the former riverbed of Turia. Turia's course was diverted to prevent constant flooding in the city after a devastating flood in 1957. Turia Gardens is now a large urban park, running nine kilometers of green space through the city, with footpaths, leisure, and sports areas, making the area very popular with locals and tourists alike. We selected the bridge railing of Puente del Ángel Custodio as the first model to complete based on the instructions in Figures 5 and 6.

During the workshop, most students could complete the design of one column, and some could also apply the loop to generate a series of columns. Just a few students had enough time to connect the columns below and top.

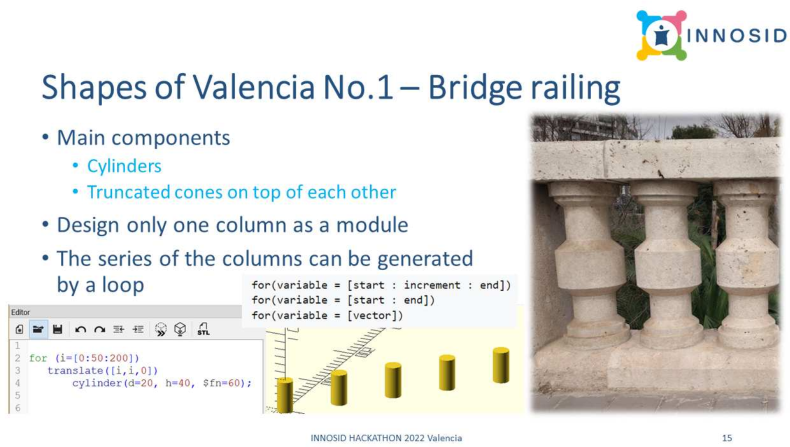


Figure 5. This slide includes a photo of the bridge railing and instructions on the main components of one item. Syntax forms of for loops and a sample code snippet about their usage are also available to help.

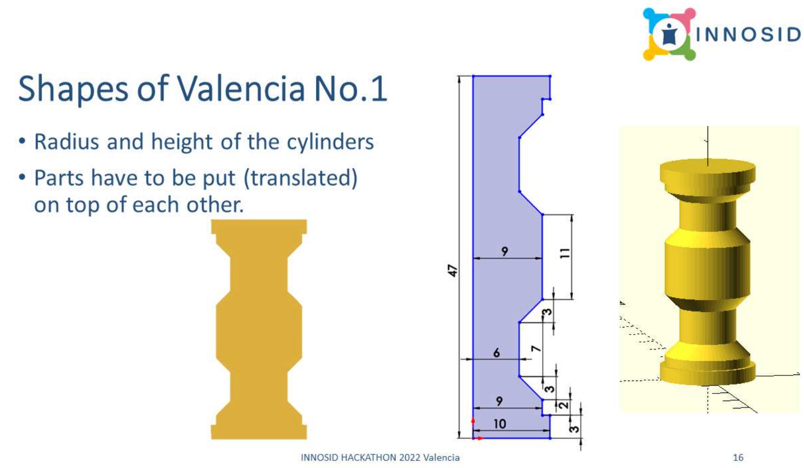


Figure 6. The second slide belonging to this task contains a CAD drawing with the dimensions and an image captured in OpenSCAD of the rendered item and its cross-section.

4.1.1. ChatGPT

First, the total free version was tested, where images cannot be used to explain the shape of the desired solid. To generate the script of one bridge rail column, we provided the list of the polygon vertices and asked ChatGPT to create a solid by rotating the polygon around the y-axis. The result was perfect, although the task was not complicated. Only the function *rotate_extrude()* should have been applied for the polygon created from the points.

In the next phase, we wanted to benefit from the visual information of the slides, too. Thus, we started to use the advanced GPT-4o model. First, the screenshot (right picture in Figure 6) was uploaded into the ChatGPT to generate the model. The result (left image in Figure 7) can be regarded as a good draft version, which was refined after a prompt: “Check the image again. The solid consists of more parts.” (right image in Figure 7). Surprisingly, on the outmost level of the module, the function *difference()* was used without defining what to remove from the union of the set of cylinders. Note that in OpenSCAD (truncated), cones can be generated using the function *cylinder()* with different parametrization. We can claim that the result was similar to the required model, and after minor changes in the code, we got the appropriate model. ChatGPT also suggested: “You can fine-tune the dimensions and radii to better match the exact proportions of your model”.

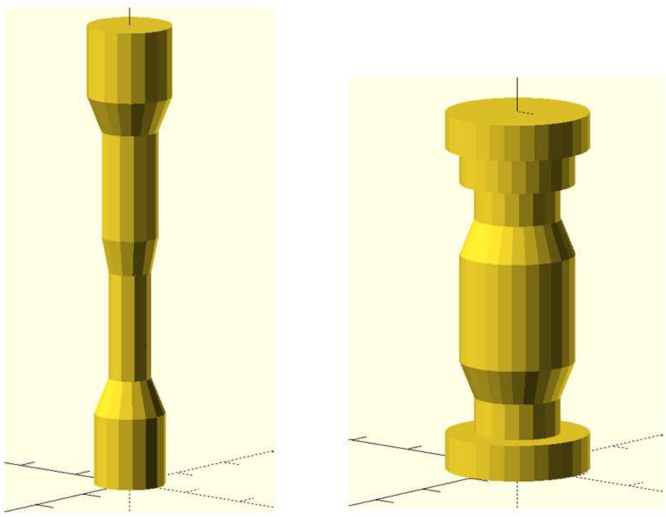


Figure 7. The models generated by ChatGPT based on the screenshot (located on the right-hand side of Figure 6). The left model is the first version, while code for the right-hand side column was proposed after a new prompt.

Since the second slide has a CAD drawing with the exact dimensions (middle picture in Figure 6), there were some students who decided to generate the column by rotating a polygon around the y-axis. This motivated us to ask AI to use the CAD drawing to generate a surface of rotation. As mentioned before, GPT-4o model can interpret not only text but images and videos. The left model in Figure 8 was created after the first prompt. ChatGPT defined a module that determined a polygon consisting of 16 points (whose positions were only partly correct). This polygon was the input shape for the *rotate_extrude()* function with an angle of 360 degrees. After the second prompt ("Please check the sketch again. The shape is not correct."), the model tended to be more similar to the desired one (see the middle solid in Figure 8). It is noticeable that there are no truncated cones at all, but the number of polygon points increased to 28. After a new prompt, the model became worse: it still only had cylinders and was missing the upper part (see the right solid in Figure 8).

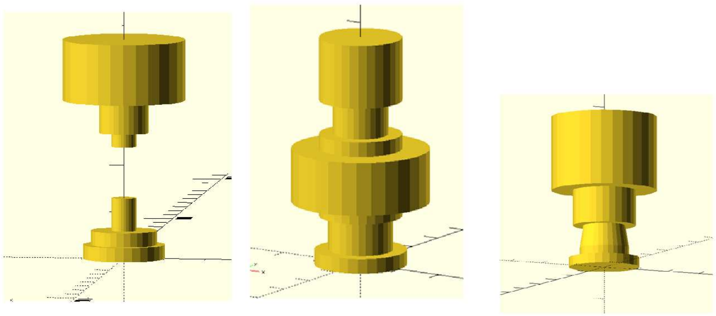


Figure 8. The models generated by ChatGPT based on the CAD drawing (located in the middle of Figure 6). The left model was proposed first, then the middle one after a second prompt. Since the middle model still consists of only cylinders, another prompt was asked, which resulted in the right model that is worse than the previous one.

If we are familiar with the basics of OpenSCAD syntax, we can alter the script to get the right model. All we have to do is insert some points into the list of polygon vertices and amend the coordinates if it is necessary.

This task cannot be completed without any OpenSCAD knowledge even if we use the current latest language model, GPT-4o.

4.1.2. Gemini

We followed the same scenario when testing Gemini; thus, no visual information was used at first. Gemini received the list of the polygon vertices and was asked to create a solid by rotating the polygon around the y-axis. Exactly the same prompt was used as in the case of ChatGPT. The proposed script and all the others, after providing further prompts to fix the syntax errors, could not be compiled in OpenSCAD. The script introduced variables to hold 2D and 3D shapes, which resulted in syntax errors. Our next prompt suggested using modules instead of variables. It is important to note that variables in OpenSCAD are set at compile-time, not run-time. Although the new script contained one module, it still had a variable preventing it from being compiled. At this stage, there was no more sense in enforcing the fixing of the code; it was faster and simpler to correct manually. Gemini does not insert comments into the script but gives an attempt to explain the recommended code. The final prompt was: "Do not use variables at all, but modules.", that resulted in the script in Figure 9, which still does not run.

In the next phase, the screenshot was first used as visual information to force Gemini Advanced to generate the OpenSCAD code of a railing column. As Figure 10 shows, the model is very simplified in contrast to the code, which is very complex. We can state that these 38 lines of code (containing seven lines of comments) are too much effort for these two concentric cylinders. The second ("Can you check the image again? The solid is more complicated.") and third prompts ("Try to create it from cylinders and truncated cones with different radii.") resulted in the same model visible right in Figure 10. At this point, there was no point in continuing prompting or trying to edit the code manually, which was rather complicated compared to the complexity of the solid.

```
module create_solid(points) {
  rotate_extrude(angle=360) polygon(points);
}

create_solid([[0, 0], [10, 0], [10, 3], [9, 3], [9, 5],
     [6, 8], [6, 15], [9, 18], [9, 29], [6, 32], [6, 39
    ], [9, 42], [9, 44], [10, 44], [10, 47], [0, 47], [
    0, 0]]);
```

Figure 9. The wrong script that Gemini generated after the prompt suggested using only modules instead of variables. The code directly passes the points to the function *polygon()* that is nested into *rotate_extrude()* function.

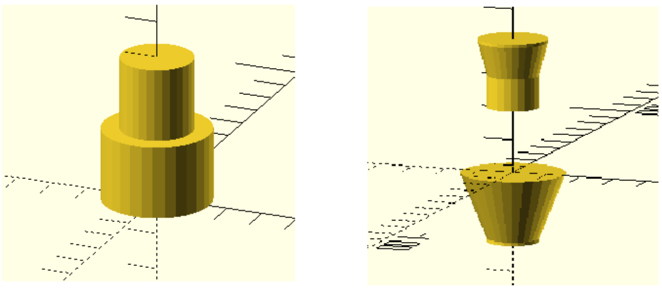
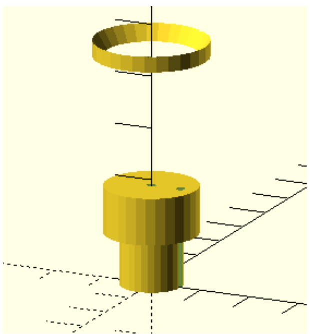


Figure 10. The solids designed by Gemini Advanced based on the OpenSCAD screenshot (right in Figure 6). Code for the left model was generated after the first prompt. The model on the right-hand side was rendered based on the code after the second and third prompts.

The experiment continued using the CAD drawing as a visual prompt for Gemini Advanced. The generated code was well-segmented: ten variables (note that ten numbers are visible in the drawing but did not match the variable values precisely) were introduced and used to create the list of vertices for the polygon to be revolved. Altogether, Gemini Advanced detected eleven vertices instead of 16, but their coordinates were not computed correctly, which resulted in an inappropriate solid (see Figure 11). The following prompts were about enforcing it to interpret the drawing more precisely (no change in the shape), taking the horizontal symmetry into account (new code with several syntax errors). Overall, we failed to use Gemini Advanced in this task. We have to mention at the same time that the first code version was capable of editing manually to have the desired OpenSCAD code.



```
// Dimensions (based on the image)
total_height = 47;
width = 11;
step_1_height = 10;
step_1_width = 6;
step_2_height = 9;
step_2_width = 9;
step_3_height = 11;
notch_depth = 3;
notch_width = 2;
fillet_radius = 1;  // Optional: Add
```

Figure 11. The model generated by Gemini Advanced based on the CAD drawing (right in Figure 6), and the introduced variables from the script. Inappropriate calculations resulted in a self-intersecting polygon that caused the two disjoint parts after rotating.

4.2. Stone Ornament

While wandering in Valencia, you can see a particular solid used to decorate the top of barriers along the Turia Gardens' walls (right color photo in Figure 12). The form of this ornament is exciting, and its design requires some knowledge of geometry. That was why we selected this ornament's form as a task in the workshop. The students received the instructions on two slides (see Figures 12 and 13), where beyond the screenshot of the model, detailed help in text and numbers was available. The solid can be divided into three main parts, and only the middle one is complicated. To better understand how it can be designed, the second slide shows a screenshot taken in TinkerCAD (an online, free 3D modeling software). The orientation of the four ellipse-based cylinders is visible, and additional information on the necessary rotation and translation is given, too. This task was more challenging since the students (who majored in IT-related degree programs) had no advanced knowledge of geometry, but most of them could complete the design based on the slides and were satisfied with the result.

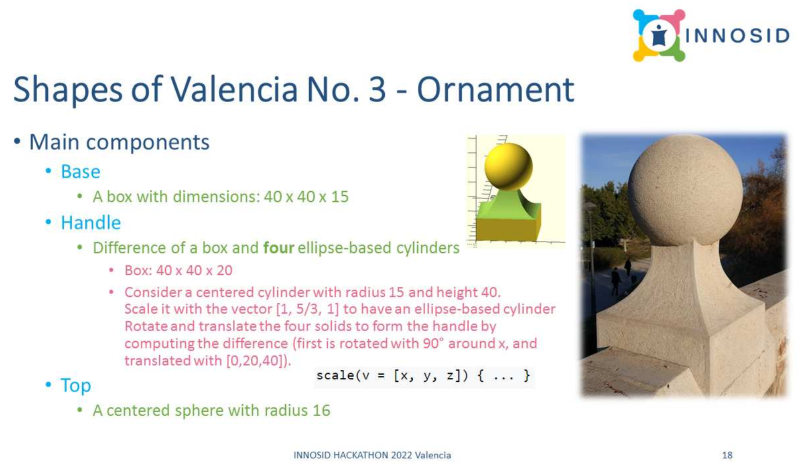


Figure 12. The three main components of the stone ornament, along with a photo and screenshot in OpenSCAD. The middle component has the most challenging shape, which justifies the need for more detailed instructions in the slide.

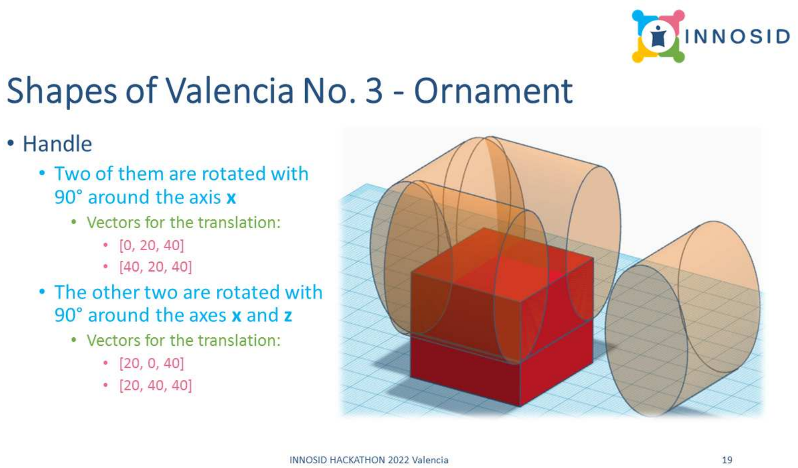


Figure 13. Detailed instruction on designing the middle part of the ornament. The right screenshot was taken in TinkerCAD to help understand the position and orientation of solids used to form the handle.

4.2.1. ChatGPT

Since the model to be designed is complex, and first, only textual information is used as prompts, we had to define the steps of modeling in detail. The following instructions were given: "Create a solid consisting of 3 parts. The first part is a box with dimensions 40 × 40 × 15. The second part is the difference of a box (40 × 40 × 20) located on the top of the first part and four ellipse-based cylinders deriving from a primitive: centered cylinder with a radius of 15 and height of 40 that is scaled with the vector [1, 5/3, 1]). The position of the cylinders: first: rotate the primitive with 90° around x, and translate it with [0, 20, 40], second: translate a copy of the first one with [40, 0, 0]. Third: rotate the primitive with 90° around x and z, and translate it with [20, 0, 40], Fourth: translate a copy of the third one with [0, 40, 0] The third part is a sphere with radius 16 placed on the second part top, but lower it a bit to intersect the handle, not only touch it". ChatGPT generated a script with modules and some comments but did not run. After specifying where OpenSCAD denoted the error, the enhanced version resulted in a solid. The issue was that ChatGPT nested two modules, which prevented the nested one from being accessed for free. Both version miscalculated the volume of translation along the z-axis for the cylinders. We also have to mention that the fragment number (parameter *$fn*) was not set for the cylinders or the sphere; the default values were used. The appropriate values were set after prompting the need for a smoother surface.

In the second turn, we uploaded the screenshot of the model taken in OpenSCAD into ChatGPT to give visual information for generating the OpenSCAD code. Based on the screenshot, it recognized three main parts whose mutual positions were acceptable (see left image in Figure 14). The middle part needed to be refined, so we used the prompt "The shape of the pedestal is more complicated. It fits exactly the base and has a curved surface from each side. Ellipsoids could be used to subtract". As the middle model (Figure 14) demonstrates, the textual information overwrote the image information since the model middle part (ChatGPT called it pedestal) shape became worse. As a last attempt, we defined the shape that could be used to form the middle part with this prompt: "Oh no, that is worse. Try to use elliptic cylinders to subtract". The result can be seen on the right in Figure 14, proving that we must provide detailed text information if we want to get the right shape. Since the model is vertically symmetric, an additional image could not provide valuable information on its shape.

Interestingly, ChatGPT assigned explicit colors to the different parts. Indeed, the screenshot taken in OpenSCAD uses not only one color to render the solid, but shading is the reason. This has no considerable significance in 3D modeling; it was just an interesting observation.

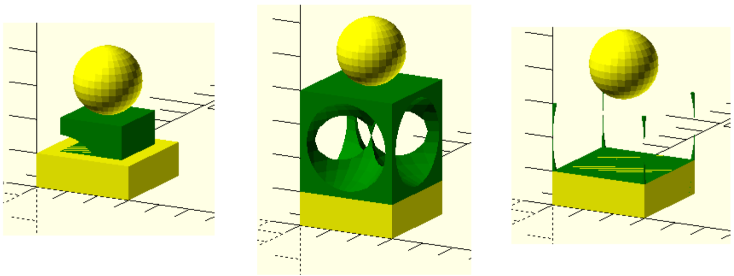


Figure 14. ChatGPT generated models based on the OpenSCAD screenshot of the stone ornament. The left image was created based only on the screenshot; in the case of the middle and right ones, textual information was also provided to refine the shape.

4.2.2. Gemini

The same detailed text instructions were given to Gemini to create the OpenSCAD script for the ornament. Gemini defined three modules for each part, just like ChatGPT, but it had issues interpreting the instructions precisely. The biggest mistake was that in the module responsible for the elliptic cylinder, the function *scale()* was misused, which resulted in no change in the dimensions. The positions of the four elliptic cylinders were calculated wrong, just like that of the sphere (see Figure 15). Gemini used the default smoothness for the cylinders and the spheres. Main errors in the code:

- wrong syntax for the function *scale()* which resulted no change in the dimensions
- the sphere was not aligned with the other parts (the default position for the cube is not centralized but for the sphere)
- wrong position for the four elliptic cylinders.

After the prompt "The cylinders are not scaled" there was no change in calling the function, and as you can see in Figure 15, the generated solid is less similar to the original ornament. There was no sense in giving further prompts at this point since it took less effort to make the necessary changes manually than to define several prompts to correct all the wrong calculations.

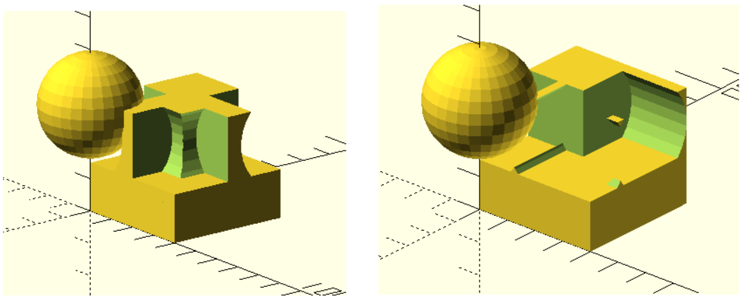


Figure 15. The solids generated by Gemini using only text prompts for the ornament. The same instructions were given as ChatGPT received in the first round. The left solid is the first model that is far from the desired one. The right-hand side model was created after a second prompt but resulted in no relevant changes.

Since we have a screenshot of the model, we tried to use it in Gemini Advanced as in the ChatGPT case. The left model in Figure 16 represents what Gemini Advanced designed based on the screenshot. Gemini Advanced detected four parts of the solid: base, cone, sphere, and cutouts. The role of the first three components is straightforward (see the right-hand side model in Figure 16) and could be matched to the parts of the ornaments. Without the cutouts, it could be considered a fair draft version, although the parts are not aligned and are not placed on top of each other. We have to notice that the most complicated middle part has become significantly simplified.

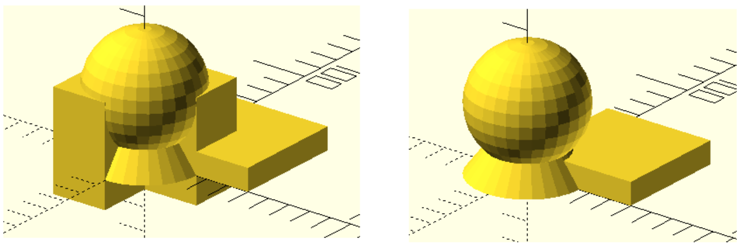


Figure 16. The model generated by Gemini Advanced based on the screenshot of the ornament model. The left model represents all four parts, while the right one does not show the cutouts (whose role is unclear).

Figure 17 shows two further versions generated after new prompts. First, we wanted a smoother surface for the sphere and a more realistic shape for the middle part (Gemini Advanced called it 'cone' in the code). The prompt was: "The sphere is not smooth enough, and the part between the sphere and base is different". The generated model (see left model in Figure 17) was improved since the sphere meets the requirement, but the shape of the middle part did not change significantly. The following prompt "The different parts are not aligned. The cone should be on the base, and the sphere should be on the cone." resulted in better alignment of the three components, but it still suffers from some inaccuracy: the sphere is disjoint, the cutouts removed some parts from the cone, and the unique shape of the middle part is only approximated. If we define the shape of the middle part as text, we can improve it, but we already tested that Gemini could not interpret our detailed description correctly.

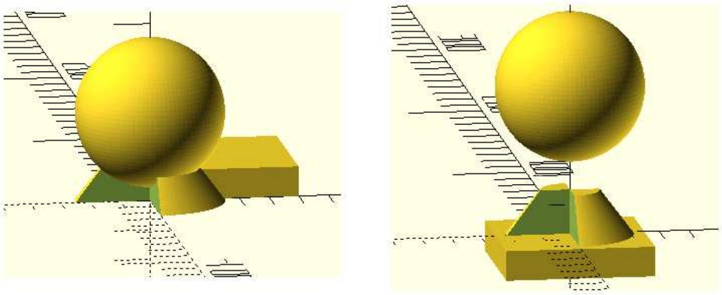


Figure 17. Models generated by Gemini Advanced after two further prompts. The first prompt resulted in the left model with a smooth circle but no improvement in the middle part. The right-hand side model has better-aligned parts but is still far from the desired shape.

To explore every possibility, we decided to check how Gemini Advanced can deal with the ornament's very detailed text prompt already used in Gemini. Unfortunately, the first offered code had the same syntax error (inappropriate usage of variables) as Gemini produced earlier. We used the prompt "Define the cylinder primitive as a module since now it has a syntax error." to overcome this issue, which already resulted in a valid code although the solid is not correct (see left image in Figure 18):

- the base cube is centered (along each axis), but its position is misused later
- positions of the four elliptic cylinders are miscalculated.

The right image in Figure 18 demonstrates the wrong positions of the four elliptic cylinders. Their actual locations, not only along the x and y but also along the z-axis, are incorrect, resulting in a wrong-shaped solid. However, the miscalculations can be fixed manually, although this requires knowledge thus we could not spare much effort with Gemini Advanced.

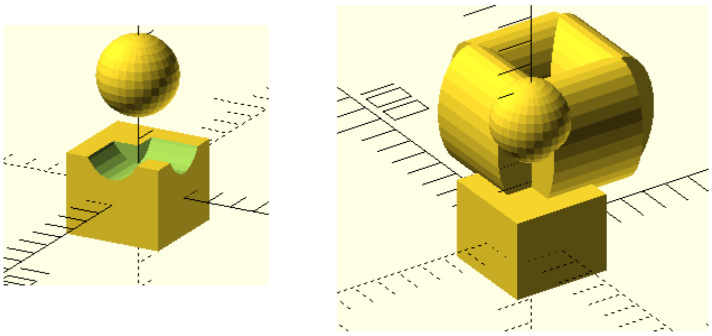


Figure 18. Gemini Advanced generated the right-hand side model after resolving the syntax errors. The left image shows the wrong positions of the cylinders caused by using the centered cube.

5. Results

We can return to our two hypotheses and assess them based on the experiments. Before doing so a table was created to provide instant visual information on the performances of the two AI chatbots (see Table 1). The assessment of the artificial intelligence's ability was stringent because we wanted to compare the AI's skills with those of the students who could successfully complete the tasks. Therefore, we established three categories:

- The model generated by the code met all the requirements.

- The proposed code generated a model that was far from the desired one and could not be used even as a sketch.
- The proposed code could be used as a draft version of the desired model. After some manual editing, the code can result in the model. The amount of needed editing was divided into two additional subcategories.

In Table 1, we used different symbols to refer to the three categories:

- (+) denotes total success
- (%) denotes relevant help where additional manual coding is necessary
- (-) denotes failure where AI could not provide help.

Results about the performance of the two tested AI tools (free and commercial) are listed in the Table 1, considering the two models separately. If the model's shape is complex, then code-based reconstruction using a picture is not feasible for either of the two tools. Without complex geometry skills, this task is challenging for humans as well. As we can see, AI has the potential to develop in the future if we want to use it in OpenSCAD to assist us in coding.

Table 1. Overview of how the ChatGPT and Gemini performed [1] while trying to use them to generate OpenSCAD scripts using textual and different visual prompts.

	Railing Column			**Stone Ornament**	
	Only Text	**Only Screenshot**	**CAD Drawing**	**Only Text**	**Only Screenshot**
ChatGPT	+	-	-	%	-
ChatGPT (GPT-4o)	+	%	%	%	-
Gemini	%	-	-	%%	-
Gemini Advanced	%	-	%	%	-

[1] Legend: +: success, -: failed, %: could be improved by editing the code manually, which requires knowledge of OpenSCAD syntax rules. The number of symbols reflects the volume of the further editing.

5.1. Testing the First Hypothesis (H1)

Hypothesis H1 is about whether the students can totally delegate their 3D modeling assignments to an AI chatbot or not. According to our experiments, the risk exists, but only if the student can precisely describe the model (which requires some modeling knowledge). It could also be achieved if images provide enough input to generate the code. Currently, the two tested AI chatbots' free versions do not support interpreting image-based information. The advanced versions are available for free as short-term trials, but they cannot even produce the desired model based only on images. The risk exists, but currently, its volume is meager.

5.2. Testing the Second Hypothesis (H2)

Hypothesis H2 concerns whether AI can provide relevant help if we design a 3D model using code-based modeling software. In this hypothesis, we do not distinguish between the prompt types. According to Table 1, in the case of both models, ChatGPT and also Gemini could provide some help in generating the code, saving some effort. Nevertheless, the chance is meager that we will not have to edit the code, which requires knowledge again. Hypothesis H2 is verified according to the results of the experiments.

6. Discussion

Emerging technologies reshape our lives in many areas. Artificial Intelligence, one of the latest achievements of the human mind, does the same and shows its impact on education, too. Many lecturers and educators at every level try to follow what is happening in the field of AI since it may have several roles in education, too. There are also some concerns, like the possibilities of delegating homework and assignments to AI tools, preventing the students from doing their own work. In this study, our objective was to examine the possible effects of introducing a code-based 3D modeling software into the curricula of our course *3D printing and modeling* at the University of Debrecen.

The key findings from the study are as follows. Teaching a code-based 3D modeling software does not increase the possibility of delegating the students' work to AI. In our experiment, we considered only single solids since the assignments to be completed by the end of the semester have to be more complicated. In some cases, AI can save some coding but cannot complete the whole code without further editing. Image-based prompts could be very convenient, but AI tools must be developed.

We conducted a test on two chatbots using their current models. It is uncertain how much they will develop in the future or when the results of this study will become outdated. These uncertainties can be seen as limitations of the study and the technology being tested. We believe that AI can be introduced in the classroom; it can save some effort but cannot totally replace the skills (either geometry or coding). Plagiarism is still a more significant threat when assessing student performance.

There are additional concerns related to applying AI in education. Institutions must introduce (some of them may have already done) new regulations about allowing or blocking its usage. Prohibition is often counterproductive, and we have no appropriate safe tool to check whether AI was used. With a more permissive regulation, a big question is how to ensure equal access to AI tools. The best, most innovative models are not free of charge. The students' financial background should not affect the range of available tools. However, someone has to pay for the development of the models. Further challenges of using AI such as critical exploration whether ChatGPT and related AI technologies serve as assets or disadvantages in education, shedding light on the unavoidable challenges encountered during their incorporation are discussed in [27].

In the future, our research will focus on testing other AI models (such as Grok-2, LLaMa-3.2, Claude Sonnet), and AI's applicability in other fields of 3D modeling and testing, including its straightforward application possibilities. With deploying image and voice capabilities of the models, they should be involved into the testing to cover all the modality of the tools. It could be interesting to test different prompting technologies with involvement students too.

7. Conclusions

Artificial intelligence tools cannot be omitted in any field of life. What we could not imagine yesterday can be real tomorrow. We must periodically reevaluate and redesign our teaching material and testing methods for students' skills in education. This study defined two hypotheses about the possible applicability of AI in code-based 3D modeling.

The first hypothesis (*Introducing code-based 3D modeling software (such as OpenSCAD) for the students has the risk that they can generate the codes using only AI chatbots.*) could not be verified thoroughly. As the experiment showed, the tested AI models in their current status can provide some help but cannot take the burden of work from the students. Thus, this hypothesis is only partially verified. Further research could give a more nuanced picture, while we also have to monitor the development of the AI tools.

The second hypothesis (*Currently available AI tools can provide relevant help in designing 3D models using a scripting language.*) could be verified with the experiment. Changes related to this hypothesis's validity can be expected in the future since the development of large language models does not stop.

Author Contributions: Conceptualization, M.Z. and I.P.; methodology, M.Z.; software, M.Z. and I.P.; validation, M.Z. and I.P.; investigation, M.Z.; resources, I.P.; writing—original draft preparation, M.Z. and I.P.; writing—review and editing, M.Z. and I.P.; visualization, M.Z. and I.P.; supervision, M.Z. and I.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All relevant data is included within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bai, C.; Dallasega, P.; Orzes, G.; Sarkis, J. Industry 4.0 technologies assessment: A sustainability perspective. *Int. J. Prod. Econ.* **2020**, *229*, 107776. [CrossRef]
2. Barata, J.; Kayser, I. Industry 5.0—Past, Present, and Near Future. *Procedia Comput. Sci.* **2023**, *219*, 778–788. [CrossRef]
3. Alhamad, I.M.; Ahmed, W.K.; Ali, H.Z.; AlJassmi, H. Chapter 6: 3D Printing Applications in Mechanical Engineering Education. In *Integrating 3D Printing into Teaching and Learning*; Ali, N., Khine, M., Eds.; Brill: Leiden, The Netherlands, 2020; pp. 90–131.
4. Özeren, Ö.; Özeren, E.B.; Top, S.M.; Qurraie, B.S. Learning-by-Doing using 3D printers: Digital fabrication studio experience in architectural education. *J. Eng. Res.* **2023**, *11*, 1–6. [CrossRef]
5. Li X.; Gould, T.; Zaki, R. The Impact of Artificial Intelligence on Learners and Teachers: A Mathematics Education Perspective. In *Education, Development and Intervention: Toward Participatory and Integrated Solutions*; Papadakis, S., Kalogiannakis, M., Eds.; Springer: Cham, Switzerland, 2024; pp. 133–148.
6. Laksmiwati, P.A.; Lavicza, Z.; Cahyono, A.N.; Alagic, M.; Mumcu, F. When engineering design meets STEAM education in hybrid learning environment: Teachers' innovation key through design heuristics. *Asia Pac. J. Educ.* **2024**, 1–19. [CrossRef]
7. Yunianto, W.; Cahyono, A.N.; Prodromou, T.; El-Bedewy, S.; Lavicza, Z. CT integration in STEAM learning: Fostering students' creativity by making Batik stamp pattern. *Sci. Act. Proj. Curric. Ideas STEM Classr.* **2024**, 1–27. [CrossRef]
8. Wang, F.; Huang, J.; Zheng, X.L.; Wu, J.Q.; Zhao, A.P. STEM activities for boosting pupils' computational thinking and reducing their cognitive load: Roles of argumentation scaffolding and mental rotation. *J. Res. Technol. Educ.* **2024**, 1–20. [CrossRef]
9. Kiliánová, K.; Kočková, P.; Biernatová, K.; Hurný, T.; Kostolányová, K. STEAM Labs: 3D Printing and Inclusive Education in Primary Schools. In Proceedings of 16th International Conference on Education and New Learning Technologies (EDULEARN24), Palma, Spain, 1–3 July 2024; pp. 7670–7675.
10. Colorado, H.A.; Mendoza, D.E.; Valencia, F.L.; Perdomo, J.M. Production of Teaching Materials Through 3D Printing as Support for Educational Processes Related to the Sciences, Heritage, and Culture. In *TMS 2020 149th Annual Meeting & Exhibition Supplemental Proceedings*; The Minerals, Metals & Materials Series; Springer, Cham, Switzerland, 2020; pp. 257–263. [CrossRef]
11. Babic, J.; Car, Z.; Gace, I.; Lovrek, I.; Podobnik, V.; Topolovac, I.; Vdovic, H.; Zilak, M.; Gourves, A.; Sable, C.; et al. *Analysis of Emerging Technologies for Improving Social Inclusion of People with Disabilities*, 1st ed.; University of Zagreb Faculty of Electrical Engineering and Computing: Zagreb, Croatia, 2020; 168p, ISBN 9789531842624. Available online: http://sociallab.fer.hr/wordpress/wp-content/uploads/2021/02/INNOSID-Report-O1.pdf (accessed on 31 July 2024).
12. Eldebeky, S.M. High School Students' Experience of a 3D Printing Station at a Bilingual School Makerspace in Kuwait. *Stud. Technol. Enhanc. Learn.* **2021**, *1*, 371–386. [CrossRef]
13. PrusaLab—Hi-Tech Prototyping Workshop from Prusa Research. Available online: https://prusalab.cz/en/ (accessed on 31 July 2024).
14. Free File Converter Online—AnyConv. Available online: https://anyconv.com/ (accessed on 30 September 2024).
15. Transform Text and 2D Images into 3D Assets with Generative AI for Free—Alpha3D. Available online: https://www.alpha3d.io// (accessed on 30 September 2024).
16. Design with Bubbles 2D to 3D in Seconds. Available online: https://www.designwithbubbles.com/ (accessed on 30 September 2024).
17. Meshy—Free 3D AI Model Generator. Available online: https://www.meshy.ai/ (accessed on 30 September 2024).
18. Jiayuan, L.; Xinyi, S.; Jeanhun, C. A Study on Process of Creating 3D Models Using the Application of Artificial Intelligence Technology. *Int. J. Adv. Cult. Technol.* **2023**, *11*, 346–351. [CrossRef]
19. Papp, I.; Tornai, R.; Zichar, M. What 3D technologies can bring to education: The impacts of acquiring a 3D printer. In Proceedings of 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Wroclaw, Poland, 16–18 October 2016; pp. 257–261.
20. TinkerCAD—From Mind to Design in Minutes. Available online: https://www.tinkercad.com (accessed on 31 July 2024).
21. SOLIDWORKS Community Download Instructions. Available online: https://www.solidworks.com/support/community-download#no-back (accessed on 31 July 2024).
22. Onwubolu, G.C. *Introduction to SolidWorks: A Comprehensive Guide with Applications in 3D Printing*, 1st ed.; Taylor & Francis Group, CRC Press: Boca Raton, FL, USA, 2017; p. 1193.
23. OpenSCAD—The Programmers Solid 3D CAD Modeller. Available online: https://openscad.org/ (accessed on 31 July 2024).
24. Solutions Based on Emerging Technologies for Improving Social Inclusion of People with Disabilities. Available online: https://sociallab.fer.hr/innosid/ (accessed on 31 July 2024).
25. ChatGPT | OpenAI. Available online: https://openai.com/chatgpt/ (accessed on 31 July 2024).
26. Gemini—Chat to Supercharge Your Ideas. Available online: https://gemini.google.com (accessed on 31 July 2024).
27. Vaughan, G.; Kovács, Á.; Szűts, Z. The educational challenges of ChatGPT. *Ann. Math. Informat.* **2023**, *58* 170–181. [CrossRef]