Silesian
University
of Technology

FINAL PROJECT

Desktop application supporting learning of stereometry

Adrian BEDNORZ
Student identification number: ab306018

Programme: Informatics
Specialisation: Computer Graphics and Software

SUPERVISOR
dr inż. Alina Momot
DEPARTMENT Katedra Informatyki Stosowanej
Faculty of Automatic Control, Electronics and Computer Science

Gliwice 2026

Thesis title

Desktop application supporting learning of stereometry

Abstract

The aim of the work is to design and implement a desktop application that supports children in learning geometry. The user of the application will have several types of tasks to choose from, in which it will be necessary to calculate the areas and volumes of three-dimensional geometric figures. There will also be an educational component where it will be possible to visualize the changes in the areas and volumes of the figures as the parameters describing the figure are altered.

Key words

application, education, geometry, stereometry

Tytuł pracy

Aplikacja na komputer stacjonarny wspomagająca naukę stereometrii

Streszczenie

Celem pracy jest zaprojektowanie i wdrożenie aplikacji desktopowej wspomagającej dzieci w nauce geometrii. Użytkownik aplikacji będzie miał do wyboru kilka typów zadań, w których będzie musiał obliczyć pola i objętości trójwymiarowych figur geometrycznych. Aplikacja będzie również zawierać część edukacyjną, umożliwiającą wizualizację zmian pól i objętości figur wraz ze zmianą parametrów opisujących figurę.

Słowa kluczowe

aplikacja, edukacja, geometria, stereometria

Contents

Chapter 1

Introduction

1.1 Introduction into the problem domain

With the development of technology, education changes from classical approaches to ones using tools and methods previously unavailable. As computers become more accessible and widespread, schools introduce more digital ways of teaching students. One of the goals of educational applications is to aid teachers in conveing knowledge more effectively than using classical methods. A well-designed educational application should be able to provide all the tools necessary to teach students a certain subject. It may also be installed on the student's personal computer, allowing them to study and practice efficiently even outside of school.

The thesis is concerned with applications which allow the user to gain and improve their skills in the field of stereometry. Stereometry (or solid geometry) is geometry in three-dimensional space. Stereometry is concerned with the measurement of surface areas and volumes of solid figures, such as polyhedrons, spheres, cyllinders and cones.

1.2 Objective and scope of the thesis

Learning stereometry may prove difficult to a student without adequate tools. There are several tools to learn stereometry without the assistance of a computer application. These include:

Blackboard and chalk Offer only static representations of figures which are difficult to draw precisely. The drawings are usually only wireframes and do not have shading. This might make it hard to visualize what the drawing is meant to represent. Drawing the figures is also time-consuming.

Printed textbooks Include shading, but are still limited to static images which can be viewed from a single angle.

Physical models Present a true three-dimesional representation which can be rotated. The dimensions are constant and not every figure might be available as a physical model.

The objective of the thesis is to design and implement a desktop application which helps the user learn stereometry.

Nowadays, web applications are more popular than desktop applications. Despite that, desktop applications have some advantages:

- Desktop apps do not require an internet connection after installation, while web apps require constant connection to the server hosting the service.
- Desktop apps are faster and more responsive as no data needs to be transfered between the client computer and a remote server.
- Desktop apps give the user control over software updates - the user may use an older version if desired.

The application will allow the user to solve solid geometry tasks, consequently expanding their skills and knowledge in this field. Its target audience are children and teenagers. The appliication can be used as a didactic tool in the classroom to help the teacher in teaching stereometry. It can also be used at home by a student to practice stereometry outside the classroom. The application will feature tasks. A task will contain one or more three-dimensional figures. The user's goal is to calculate the total area and volume of said figures. After solving a task incorrectly, the user will receive a tip, to help them solve the task successfully. A database containing all mathematical formulas, as well as tips related to the tasks will be accessible. The user will have the possibility to change the task data and see how the shapes change dynamically. An important part of the application is its customizability - the user should be able to tweak the settings of the application as well as each task such that they get the best experience tailored to their needs.

The application will be built using the Godot game engine. While designed primarily with video games in mind, it can be used to create other types of software. The application requires a modern graphical user interface (GUI) and 3D rendering capabilites. Godot provides a suite of tools which simplify working with both of those systems. Other solutions like Flutter and Electron were considered but ultimately rejected. They provide a wide array of tools for developing desktop applications but do not have first-class 3D rendering support which would make developing the application more difficult.

1.3 Short description of chapters

- The second chapter analyses the problem and presents already existing solutions.

- The third chapter states the functional and nonfuctional requirements, describes the use cases and presents the tools used in the project's development.
- The fourth chapter serves as the user manual. It specifies the sytem requirements and provides instructions on how to run and use the application.
- The fifth chapter is focused on the implementation of the project. It explains the architecture of the application.
- The sixth chapter **TODO**.
- The seventh chapter concludes the thesis. It compares the achieved results to the objectives of the thesis and explores prospects of the project's further development.

Chapter 2

Problem analysis

2.1 Mathematical description of selected solids

2.1.1 Prism

A prism is a polyhedron consisting of two congruent n-sided polygon bases and n joining faces. Every joining face is a parallelogram. A prism is called after its base, e.g. a prism with a hexagonal base is called a hexagonal prism. An n-gonal prism has $n + 2$ faces, $3n$ edges, and $2n$ vertices.

Special cases

An *oblique prism* is a prism whose joining faces are not perpendicular to the bases. A prism whose whose joining faces are perpendicular to the bases is a *right prism*.

A right prism with a rectangular base is also called a *cuboid*. A right prism with a square base is also called a *square cuboid*. A square cuboid whose height is equal to the side length of its base is a *cube*.

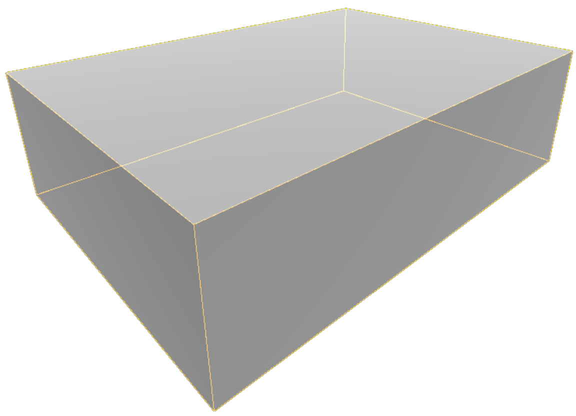

Figure 2.1: A right prism with a rectangular base.

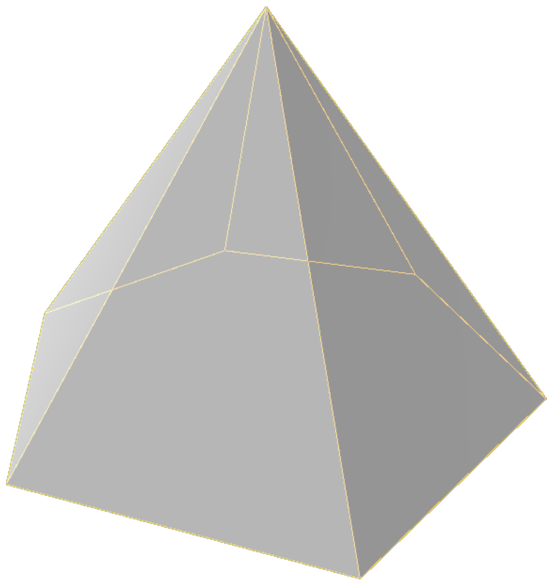

Figure 2.2: A right pyramid with a hexagonal base.

Properties

Volume The volume of a prism is the product of the base area and the height. The formula for the volume is:

$$V = Bh, \tag{2.1}$$

where B is the base area and h is the height.

Surface area The surface area of a prism is the sum of all its faces' areas. The formula for the surface area is:

$$A = 2B + Ph, \tag{2.2}$$

where B is the base area, P the perimeter, and h the height.

2.1.2 Pyramid

A pyramid is a polyhedron consisiting of one polygonal base face connected to a point, called the apex. Each of the base's edges is connected to the apex, forming a triangle, called a lateral face. A pyramid with an n-gonal base has $n + 1$ faces, $2n$ edges, and $n + 1$ vertices.

Special cases

A pyramid is called a *regular pyramid* when its base is a regular polygon. The definition of a *right pyramid* varies between sources. Some sources define it as a regular pyramid whose height falls on the center of the base.[4] According to other sources, a right pyramid is a pyramid, whose height falls on the center of a circle circumscribed about the pyramid's base.[8] A pyramid with a triangular base is also called a *tetrahedron*.

Properties

Volume The volume of a pyramid is one third of the product of the base area and the height. The formula for the volume is:

$$V = \frac{1}{3}Bh, \tag{2.3}$$

where B is the base area and h is the height.

Surface area The surface area of a pyramid is the sum of the area of the base and the area of the lateral faces. For a regular right pyramid with an n-gonal base and height h, the surface area can be calculated as follows:

Each regular n-gon can be divided into n congruent triangles. The base of the triangle is the side s of the pyramid. The triangle's height h_B is calculated as:

$$h_B = \frac{1}{2}s\tan(\frac{n-2}{n}90°) \tag{2.4}$$

Then, the polygon base area can be calculated by multiplying the area of one triangle by the number of sides:

$$\begin{aligned} A_B &= \frac{sh_B}{2}n \\ &= \frac{1}{2}s(\frac{1}{2}s\tan(\frac{n-2}{n}90°))n \\ &= \frac{1}{4}s^2\tan(\frac{n-2}{n}90°)n \end{aligned} \tag{2.5}$$

Height of lateral triangles can be obtained by using the Pythagorean theorem:

$$h_L = \sqrt{{h_B}^2 + h^2} \tag{2.6}$$

The lateral area is calculated in a similar way to the base area:

$$\begin{aligned} A_L &= \frac{sh_L}{2}n \\ &= \frac{s\sqrt{{h_B}^2 + h^2}}{2}n \end{aligned} \tag{2.7}$$

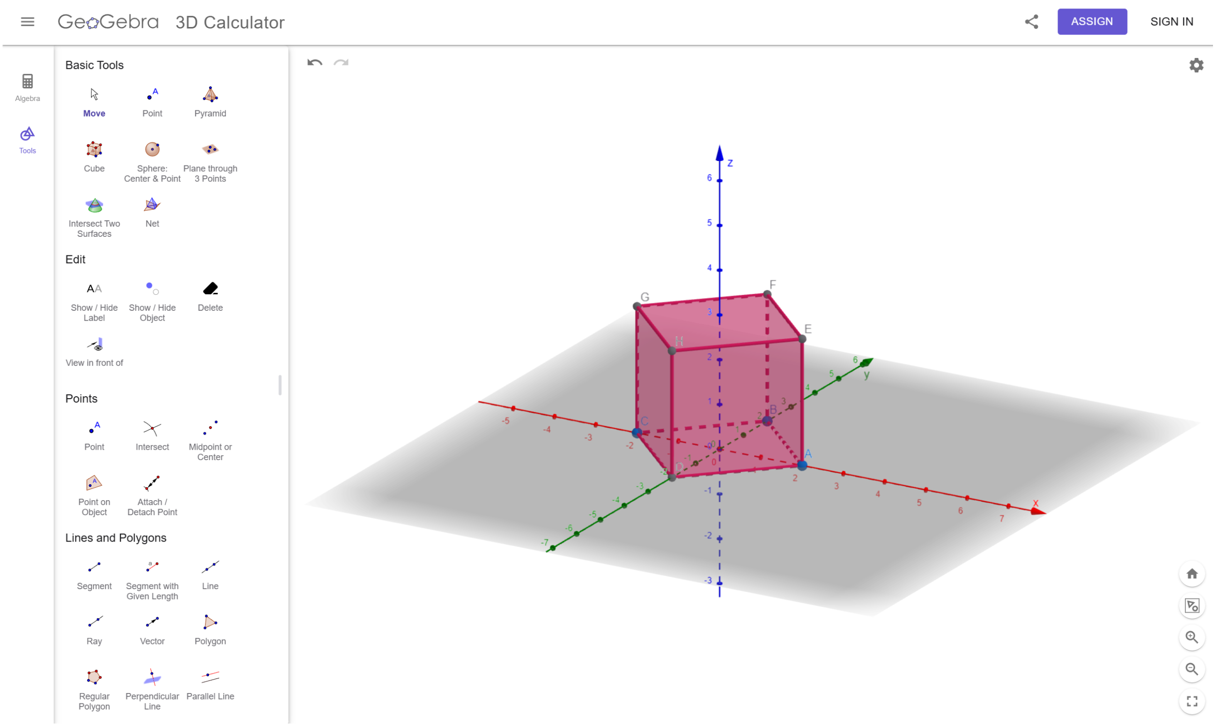

Figure 2.3: GeoGebra 3D Calculator.

Finally, the total area is the sum of the base area and the lateral area:

$$\begin{aligned} A &= A_B + A_L \\ &= \frac{1}{4}s^2\tan(\frac{n-2}{n}90°)n + \frac{s\sqrt{{h_B}^2+h^2}}{2}n \\ &= \frac{1}{2}sn(\frac{1}{2}s\tan(\frac{n-2}{n}90°) + \sqrt{{h_B}^2+h^2}) \end{aligned} \tag{2.8}$$

2.2 Existing solutions

2.2.1 GeoGebra 3D Calculator

The 3D Calculator [3] by GeoGebra is an extensive application which provides many tools for exploring 3D geometry. It allows the user to create and manipulate solids, points, lines and polygons. It also provides tools for measuring angles, distances, areas and volumes. The 3D Calculator can be of great help when exploring stereometry, but it does not provide didactic features. It is not suited for structured, guided learning. The measuring tools allow the user to calculate certain characteristics but do not teach how to do so. GeoGebra 3D Calculator does not provide support for keyboard control, the mouse being the only allowed input method. The application is only available on web and mobile platforms with no downloadable version for desktop computers.

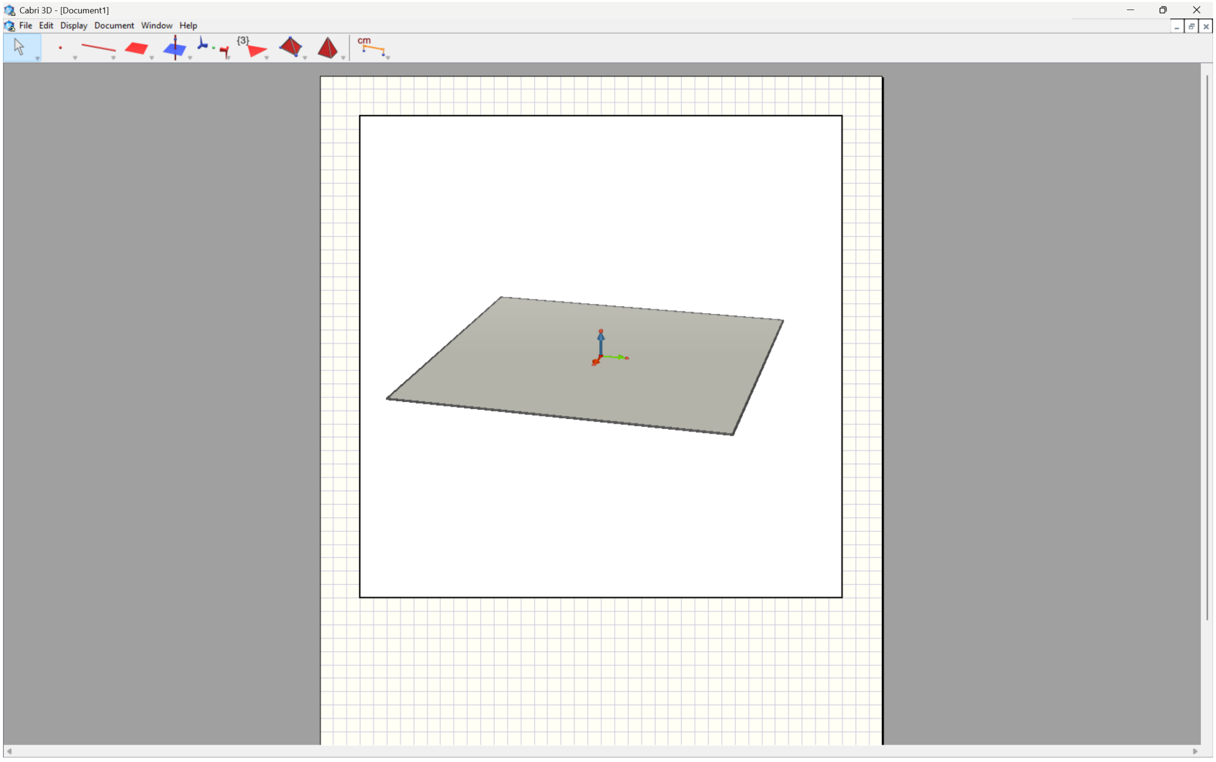

Figure 2.4: Cabri 3D.

2.2.2 Cabri 3D and Cabri Express

Cabri 3D [1] and Cabri Express [2] are 2 applications for learning three-dimensional geometry developed by Cabri. Both applications are designed with teaching in mind. Cabri products are closed-source, which means that the software will no longer be able to receive updates after the original creators stop developing them.

Cabri 3D

Cabri 3D is an application for the Windows and Mac OS operating systems. It is commencial software which requires the purchase of a license. The application is used primarily to create documents consisting of text areas and 3D views. This makes it suitable to be used by teachers to prepare lesson materials, which can be printed on paper. It is not suited to by used by students to explore 3D geometry or solve tasks. The user inteface has an outdated look and has some bugs when running on a system with multiple displays. When running on a system with two displays, the *Tool Help* window produces visual artifacts on the other display when it is moved.

Cabri Express

Cabri Express is a general mathematical application and features tools not limited to 3D geometry. It has tools related to stereometry but it is not specialized in this field. This, combined with it being closed-source introduces a problem. If one doesn't find the desired

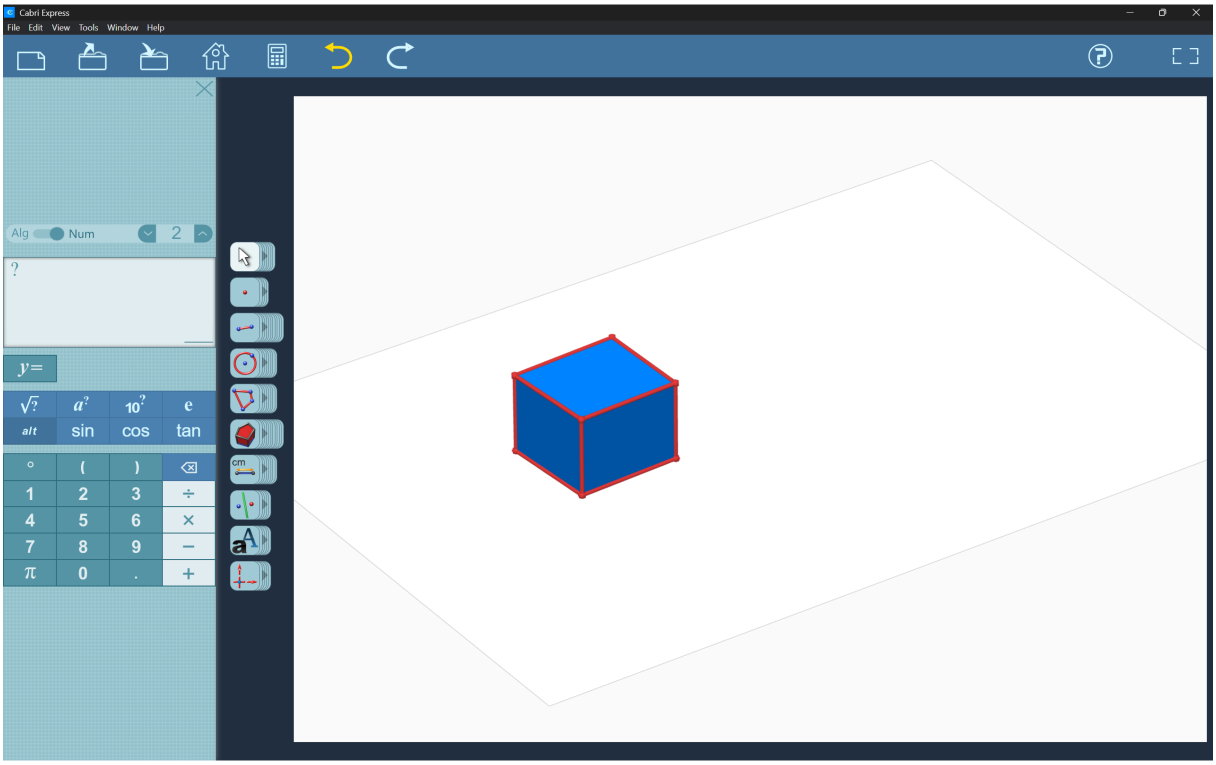

Figure 2.5: Cabri Express.

3D tool in Cabri Express, it may never be added by the developers. Learning how to use the applications may be problematic, as most of the official tutorials for the applications are in French. Moreover, there are few unofficial learning resources in other languages.

2.2.3 Shapes 3D Geometry Learning & Drawing

Shapes 3D Geometry Learning [7] and Shapes 3D Geometry Drawing [6] are two similar applications whose main feature is viewing solids. Both applications are available in a subscription model. One may also buy a perpetual license for the Shapes 3D Geometry Drawing application on the iOS platform. A free demo accessible from the browser exists for both applications. Unfortunately, the author of the thesis was unable to run the Shapes 3D Geometry Drawing demo due to an error present on the website.

Shapes 3D Geometry Learning

Shapes 3D Geometry Learning is available on all major desktop and mobile operating systems. The target group of the application is students of grades 4-6. The application allows the user to select from a set of predefined solids. When a solid is selected, the user can view its edges, faces and vertices. The solid can be unfolded it into a net. The user may also build the net themself, using the solid's faces. The net can be printed and subsequently cut out of the paper, to create a paper net. The paper net can be glued to obtain a paper version of the solid displayed in the application.

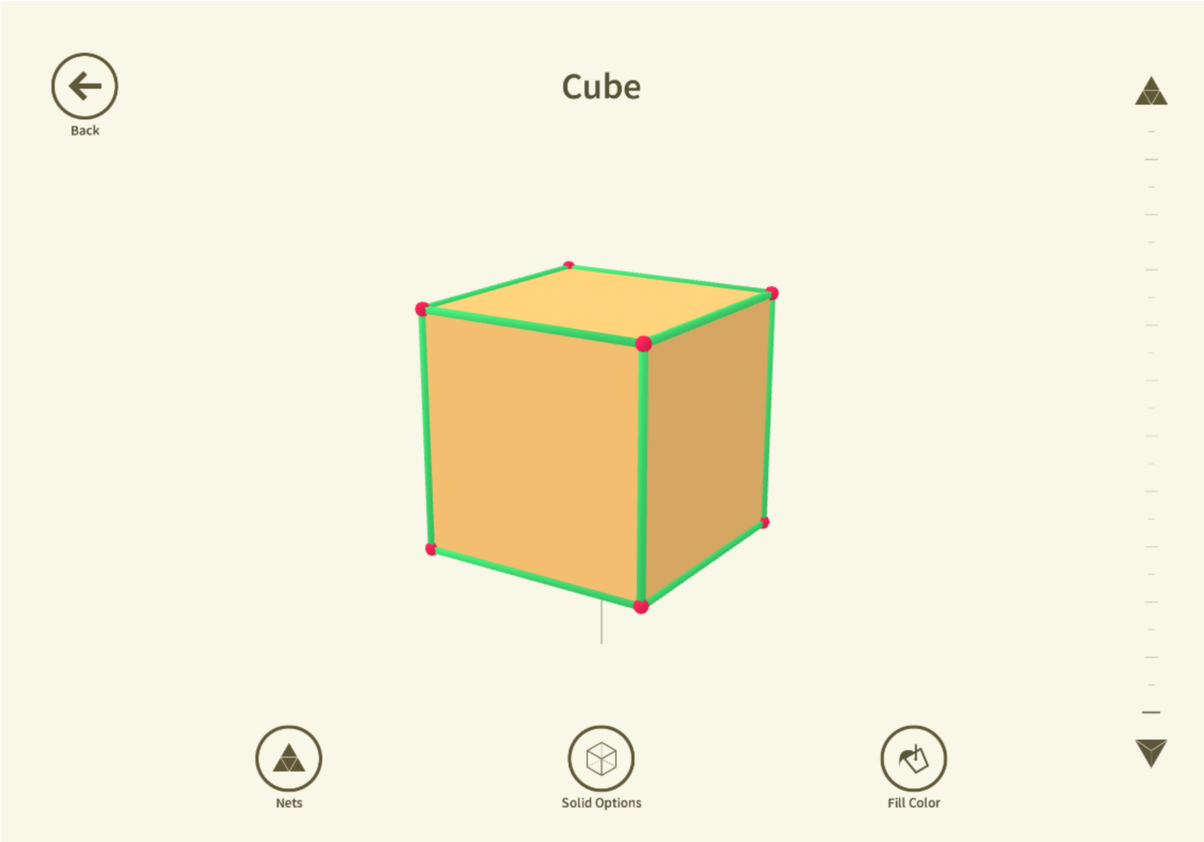

Figure 2.6: Shapes 3D Geometry Learning.

Shapes 3D Geometry Drawing

Shapes 3D Geometry Drawing is available only on the iOS operating system. The target group of the application is students of grades 7-8. It has a user interface and application structure similar to Shapes 3D Geometry Learning. The main differences is the tools available when a solid is selected. Shapes 3D Geometry Drawing allows the user to place line segments and cross sections between points on the solid's faces.

Summary of Shapes 3D applications

Both applications contain many features which help visualize and explore 3D geometry. The applications lack features related to calculation of solids' characteristics like area and volume. They also do not allow the user to create their own solid and only give access to predefined solids. The predefined solids cannot be manipulated by the user, e.g the scale of the solid cannot be changed.

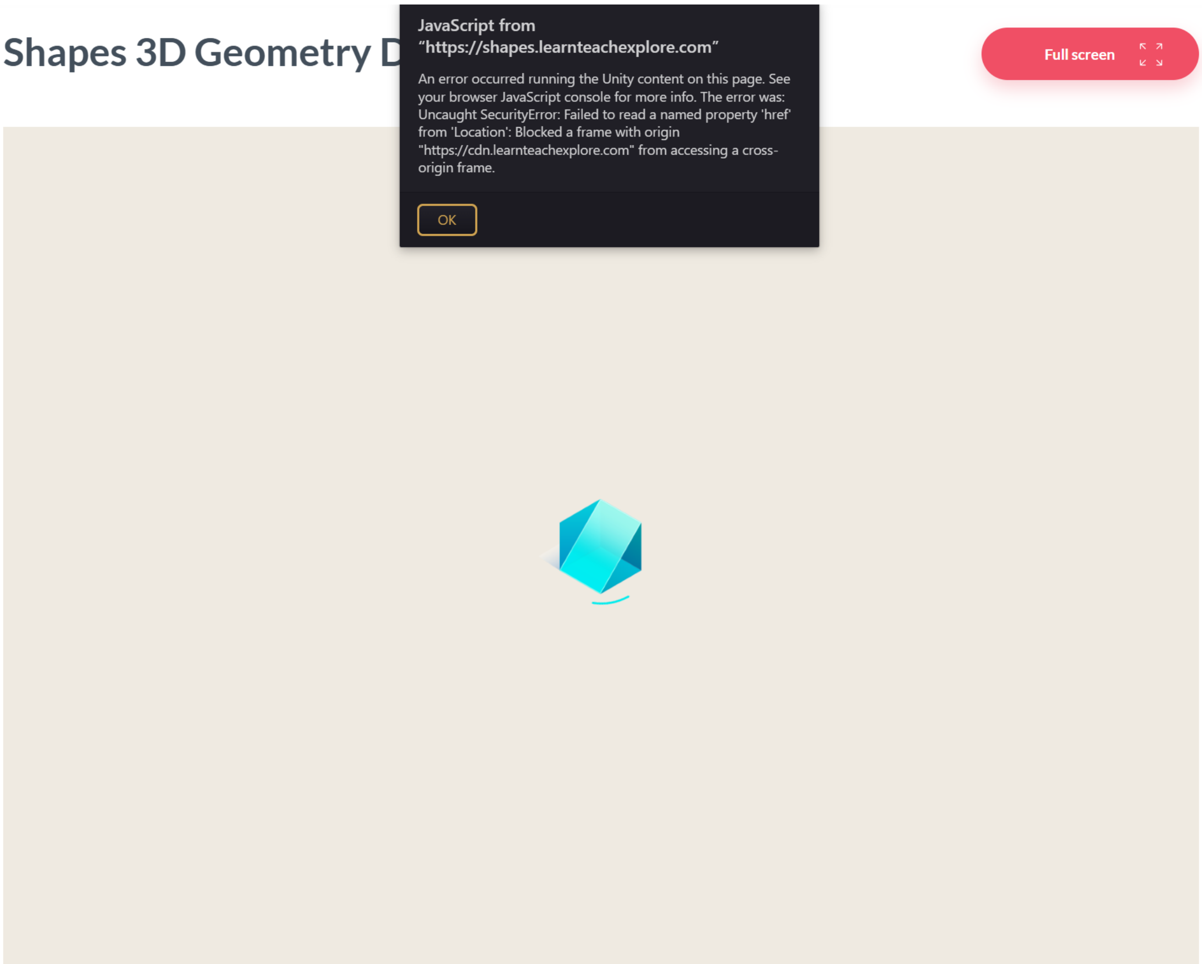

Figure 2.7: Shapes 3D Geometry Drawing. An error occurs when trying to run the demo.

Chapter 3

Requirements and tools

3.1 Functional requirements

Visualization of 3D objects The application will be able to display three-dimensional solids (e.g. cube, prism, sphere, cylinder). The camera will allow for rotation and zooming.

Inspection of solids When hovering on a solid, its vertices or edges, the object will be highlighted to visually indicate that it is selected. Upon clicking a selected object, an informational label will appear. The informational label will contain information related to the object, e.g. the position of a vertex or length of an edge.

Manipulation of solids The application will allow the user to change the parameters of tasks. The changes in parameters will be reflected in the 3D view. The application will contain a module which will allow for creation and manipulation of custom polyhedrons.

Educational support The application will contain a task module. There will be a number of predefined tasks from the field of stereometry with varying degrees of difficulty. Each task will be divided into steps. Each step will involve calculation of one portion of the problem described by the task.

User interface The application will provide an intuitive graphical user interface. The application will support mouse and keyboard interaction. User interface elements will include tooltips to help the user understand how the application works.

3.2 Non-functional requirements

Usability The user interface will be straightforward to use and understand. It will not be required for the user to be familiar with other 3D software to use the application effectively. Language within the application will be simple and use terms consistent with those use in mathematics education.

Performance The application will perform well on hardware capable of smoothly running other similar applications. There will be no noticeable stutters and delays when interacting with the application.

Reliability The application will not crash during runtime and gracefully handle invalid input.

Correctness The application will correctly calculate all characteristics of 3D objects. The user will be able to set the precision with which the results will be displayed.

Maintainability The application code will be open-source. The application will be structured in a way allowing easy future extension in the form of new tasks and features.

Security The application will not require internet access after installation. Manual editing of files used by the application will not compromise security.

Educational suitability The application will be suited for use in the classroom and at home. The provided tasks will be comparable to tasks from a school stereometry curriculum.

Chapter 4

External specification

4.1 System requirements

A desktop computer or a laptop is required the application. The application can run on the Windows, macOS and Linux operating systems. The application requires approximately 100 MB of storage space. The application was mainly tested on a laptop device with the following software and hardware configuration:

CPU	AMD Ryzen 7 5800H
GPU	NVIDIA GeForce RTX 3050
RAM	16 GB
Operating system	Windows 11

The application performed smoothly on the device. Due to limited access to other hardware, the application could not be tested on low-end devices to establish minimum system requirements accurately. The Godot Engine documentation lists the following as the minimum system requirements:[5]

CPU	x86_32 CPU with SSE2 support, x86_64 CPU with SSE4.2 support, ARMv8 CPU
GPU	Integrated graphics with full Vulkan 1.0 support, Metal 3 support (macOS) or Direct3D 12 (12_0 feature level) support (Windows)
RAM	2 GB
Operating system	Windows 10, macOS 10.15, Linux distribution released after 2018

4.2 Installation

The application does not require installation on any supported platform. It is launched directly from the appropriate executable.

The application was tested only on the Windows operating system. As such, the instructions for other operating systems may contain errors.

4.2.1 Windows

For the Windows operating system, the application is provided as a single executable file. To start the application, the user should run the **GeoApp.exe** executable.

Depending on system security settings, Windows may display a warning about an unknown publisher. In this case, the user can confirm the warning to start the application. The executable is compiled for the x86-64 architecture.

4.2.2 Linux

For the Linux operating system, the application is provided as a single executable file. The user should ensure that the file has execution permissions. This can be done through the file manager or by executing the following command in the terminal:

```
chmod +x GeoApp.x86_64
```

When the file has execution permissions, the application can be started by running the **GeoApp.x86_64** executable. The executable is compiled for the x86-64 architecture.

4.2.3 macOS

On the macOS operating system, the application is provided in a `.zip` archive containing an application bundle. First, the user should extract the provided **GeoApp.zip** file. The user may then optionally move the extracted **GeoApp.app** file to the **Applications** folder. To start the application, the user should run the **GeoApp.app** application.

Since the application is not digitally signed, a security warning may appear when the application is started for the first time. When this happens, the user can start the application by right-clicking the **GeoApp.app** file, selecting **Open** and confirming the dialog. Alternatively, the application can be allowed in the system settings under **Privacy & Security**.

4.3 User manual

The application consists of several modules, namely *Task exploration*, *Task solving* and the *Playground*. Each module provides different ways for the user to learn stereometry.

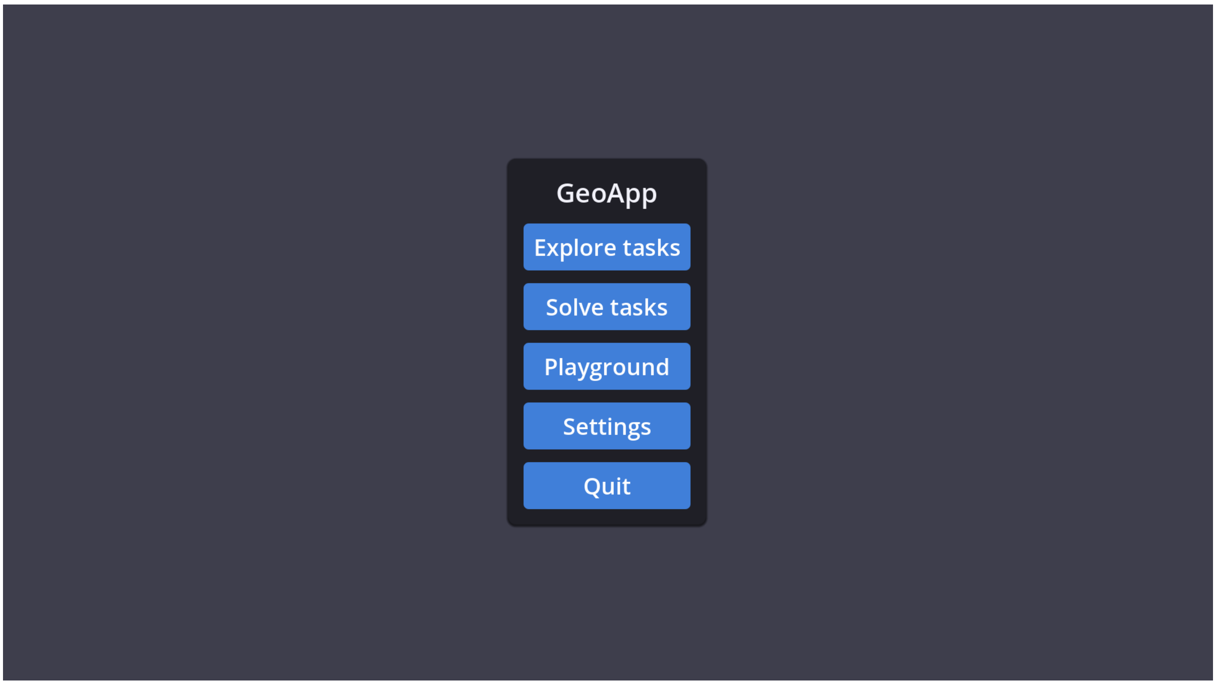

Figure 4.1: The main menu of the application.

4.3.1 Main menu

Upon starting the application, the user sees the main menu. The main menu is a central hub from which the rest of the application can be accessed. The main menu contains several button used to navigate the application.

- The *Explore tasks* button takes the user to the *Task exploration* module.
- The *Solve tasks* button takes the user to the *Task solving* module.
- The *Playground* button takes the user to the *Playground* module.
- The *Settings* button takes the user to the settings.
- The *Quit* button closes the application.

4.3.2 Task exploration

The purpose of the *Task exploration* module is to teach the user about the tasks available in the application. Ordered from left to right, the user interface features:

- A list of all task, grouped by difficulty. The tasks can also be filtered by their name.
- All data describing the task. Each parameter can be changed. All UI (user interface) elements will update dynamically when a parameter is changed.

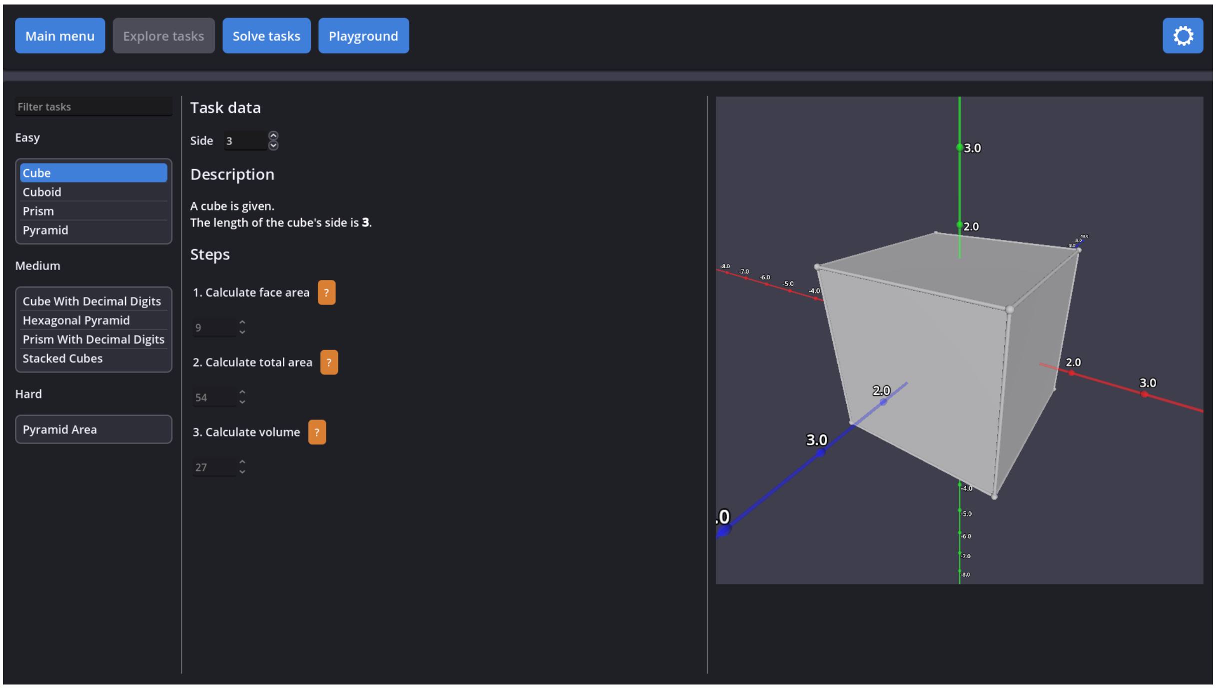

Figure 4.2: The *Task exploration* module.

- The description of the task.
- A list of all of the steps required to solve the tasks. For each step, there is:
  - The description of what should be calculated for the step.
  - A help button. The button shows a hint for its respective step when pressed.
  - A number control which displays the correct answer for its respective step.
- A view of the selected task.

4.3.3 Task solving

The purpose of the *Task solving* module is to test the user's knowledge and skills gained in the *Task exploration* module. The *Task solving* module is split into two parts.

The first part is task selection. Here, the user selects the task they wish to solve. To select a task, the user must first select a difficulty. When a difficulty is selected, all available tasks with this difficulty are shown. Then, the user should select their desired task and press the **Start** button. The **Start** button can be pressed only when a task is selected.

After pressing the **Start** button, the user is transported to the main part of the *Task solving* module.

The parameter values of the task are randomly picked from a predefined range. The range is defined on a per-parameter basis. The precision (the number of decimal digits of the value) is also defined on a per-parameter basis.

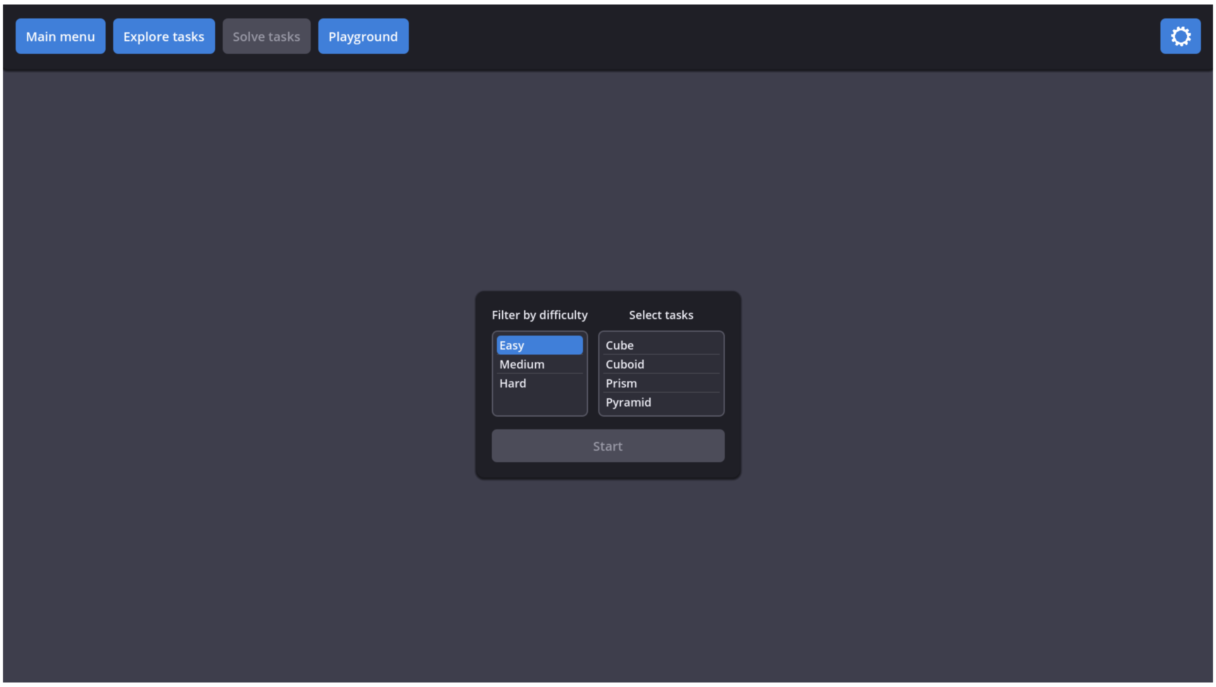

Figure 4.3: Task selection.

Inside the main part of the *Task solving* module, the user can find:

- The view of the current task.
- The description of the task.
- The step navigation buttons.
- The *step container*.
- The **Check answer** button.
- The **Get new task** button.

The *step container* is the part of the UI responsible for displaying the task's steps. The step container looks similar to the one present in the *Task exploration* module. Despite that, the two differ in several aspects.

First, the user can interact with the number controls. The number controls are where the user puts their solutions to the task. The user may edit the value only of the current step's number control. If the user enters the correct value and presses the **Check answer** button, a green checkmark symbol appears to the right side of the number control. If the user enters an incorrect value, a red X symbol appears instead.

Second, only steps up to the current step are visible. The navigation buttons are used to move between steps. The **< Prev** button moves to the previous step. The **Next >** button moves to the next step. The buttons are active only when the user is able to change

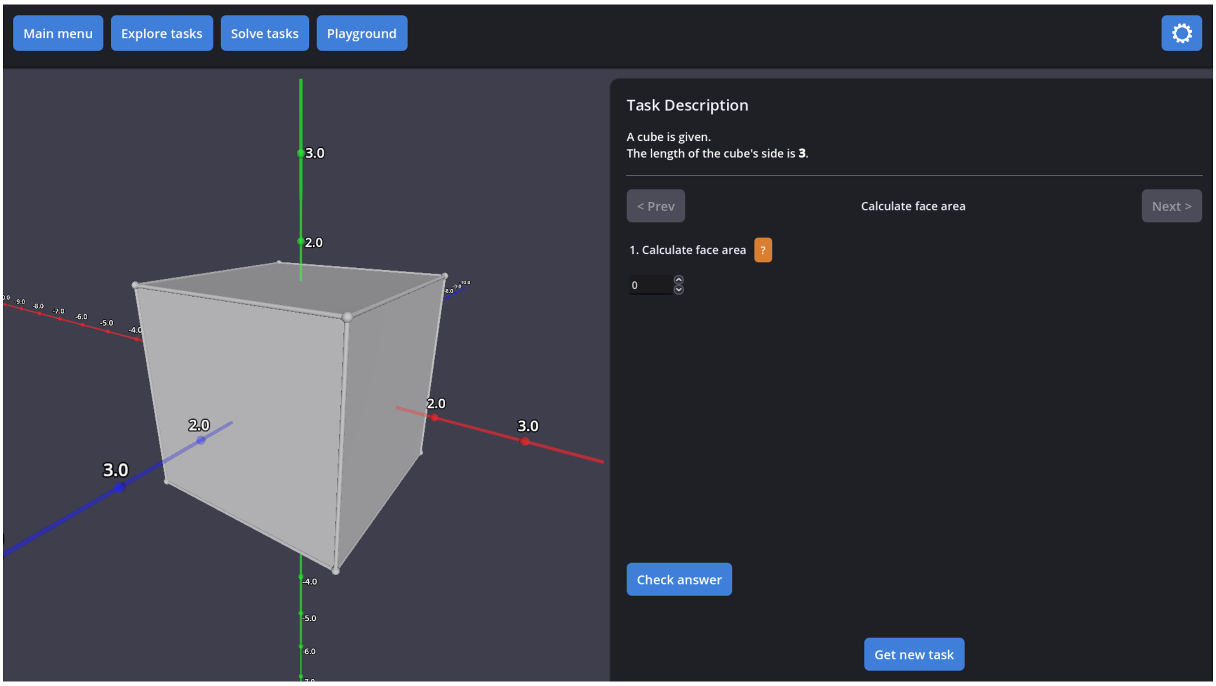

Figure 4.4: The default state of the task solving screen.

the current step. The < **Prev** button is inactive when the user is on the first step of the task. The **Next** > button is inactive when the user is on the last step of the task or the answer for the current step is incorrect.

When the user enters the correct answer for the last step of the task, the task is completed. When a task is completed, a label with the text "Task complete!" appears under the **Check answer** button.

The **Get new task** button resets the current task and provides the user with a new one with randomized values. Returning to the task selection screen and selecting the same task achieves the same goal.

4.3.4 Playground

4.3.5 Settings

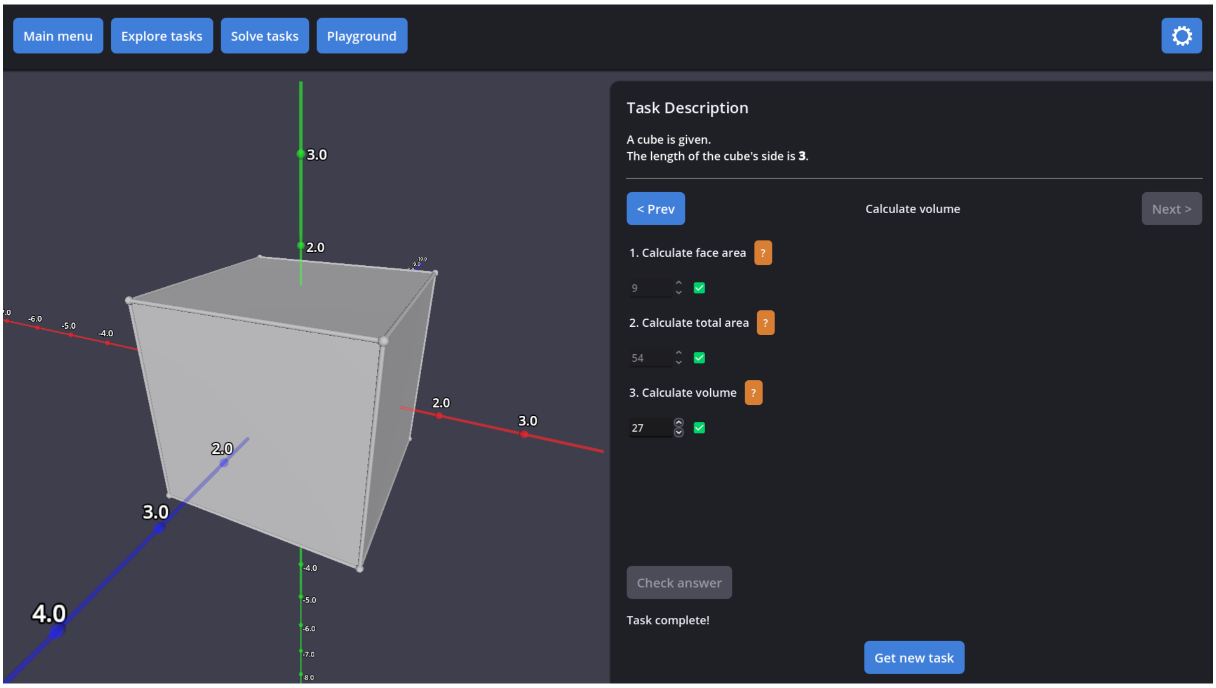

Figure 4.5: The task solving screen after the current task is completed.

Chapter 5

Internal specification

- concept of the system
- system architecture
- description of data structures (and data bases)
- components, modules, libraries, resume of important classes (if used)
- resume of important algorithms (if used)
- details of implementation of selected parts
- applied design patterns
- UML diagrams

Use special environments for inline code, eg `int a;` (package `listings`). Longer parts of code put in the figure environment, eg. code in Fig. 5.1. Very long listings–move to an appendix.

```
class test : public basic
{
    public:
      test (int a);
      friend std::ostream operator<<(std::ostream & s,
                                     const test & t);
    protected:
      int _a;

};
```

Figure 5.1: Pseudocode in `listings`.

Chapter 6

Verification and validation

- testing paradigm (eg V model)
- test cases, testing scope (full / partial)
- detected and fixed bugs
- results of experiments (optional)

Table 6.1: A caption of a table is **above** it.

	method						
			alg. 3			alg. 4, $\gamma = 2$	
ζ	alg. 1	alg. 2	$\alpha = 1.5$	$\alpha = 2$	$\alpha = 3$	$\beta = 0.1$	$\beta = -0.1$
0	8.3250	1.45305	7.5791	14.8517	20.0028	1.16396	1.1365
5	0.6111	2.27126	6.9952	13.8560	18.6064	1.18659	1.1630
10	11.6126	2.69218	6.2520	12.5202	16.8278	1.23180	1.2045
15	0.5665	2.95046	5.7753	11.4588	15.4837	1.25131	1.2614
20	15.8728	3.07225	5.3071	10.3935	13.8738	1.25307	1.2217
25	0.9791	3.19034	5.4575	9.9533	13.0721	1.27104	1.2640
30	2.0228	3.27474	5.7461	9.7164	12.2637	1.33404	1.3209
35	13.4210	3.36086	6.6735	10.0442	12.0270	1.35385	1.3059
40	13.2226	3.36420	7.7248	10.4495	12.0379	1.34919	1.2768
45	12.8445	3.47436	8.5539	10.8552	12.2773	1.42303	1.4362
50	12.9245	3.58228	9.2702	11.2183	12.3990	1.40922	1.3724

Chapter 7

Conclusions

- achieved results with regard to objectives of the thesis and requirements
- path of further development (eg functional extension . . .)
- encountered difficulties and problems

Bibliography

[1] Cabrilog. *Cabri 3D*. URL: https://cabri.com/en/enterprise/cabri-3d/index.html (visited on 12/01/2026).

[2] Cabrilog. *Cabri Express*. URL: https://cabri.com/fr/enterprise/cabri-express/ (visited on 12/01/2026).

[3] GeoGebra GmbH. *GeoGebra 3D Calculator*. URL: https://www.geogebra.org/3d (visited on 12/01/2026).

[4] S. Gottwald. *The VNR Concise Encyclopedia of Mathematics*. Springer Netherlands, 2012, p. 193. ISBN: 9789401169844. URL: https://books.google.pl/books?id=bCLfnQEACAAJ.

[5] Ariel Manzur Juan Linietsky and the Godot community. *System requirements*. URL: https://docs.godotengine.org/en/stable/about/system_requirements.html (visited on 18/01/2026).

[6] Learn Teach Explore Sp z o. o. *Shapes 3D Geometry Drawing*. URL: https://shapes.learnteachexplore.com/shapes-3d-geometry-drawing/ (visited on 12/01/2026).

[7] Learn Teach Explore Sp z o. o. *Shapes 3D Geometry Learning*. URL: https://shapes.learnteachexplore.com/shapes-3d-geometry-learning/ (visited on 12/01/2026).

[8] G. Polya. *Mathematics and Plausible Reasoning: Induction and analogy in mathematics*. Induction and Analogy in Mathematics. Princeton University Press, 1990, p. 138. ISBN: 9780691025094. URL: https://books.google.pl/books?id=-TWTcSa19jkC.

Appendices

Index of abbreviations and symbols

DNA deoxyribonucleic acid

MVC model–view–controller

N cardinality of data set

μ membership function of a fuzzy set

$\mathbb{E}$ set of edges of a graph

$\mathcal{L}$ Laplace transformation

Listings

(Put long listings here.)

```
if (_nClusters < 1)
    throw std::string ("unknown number of clusters");
if (_nIterations < 1 and _epsilon < 0)
    throw std::string ("You should set a maximal number of iteration or minimal difference -- epsilon.");
if (_nIterations > 0 and _epsilon > 0)
    throw std::string ("Both number of iterations and minimal epsilon set -- you should set either number of iterations or minimal epsilon.");
```

List of additional files in electronic submission (if applicable)

Additional files uploaded to the system include:

- source code of the application,
- test data,
- a video file showing how software or hardware developed for thesis is used,
- etc.

List of Figures

List of Tables