



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 2025



## **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

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Introduction into the problem domain . . . . .	1
1.2	Objective and scope of the thesis . . . . .	1
1.3	Short description of chapters . . . . .	2
<b>2</b>	<b>[Problem analysis]</b>	<b>5</b>
<b>3</b>	<b>Requirements and tools</b>	<b>7</b>
3.1	Functional requirements . . . . .	7
3.2	Non-functional requirements . . . . .	7
<b>4</b>	<b>External specification</b>	<b>9</b>
<b>5</b>	<b>Internal specification</b>	<b>11</b>
<b>6</b>	<b>Verification and validation</b>	<b>13</b>
<b>7</b>	<b>Conclusions</b>	<b>15</b>
	<b>Bibliography</b>	<b>17</b>
	<b>Index of abbreviations and symbols</b>	<b>21</b>
	<b>Listings</b>	<b>23</b>
	<b>List of additional files in electronic submission (if applicable)</b>	<b>25</b>
	<b>List of figures</b>	<b>27</b>
	<b>List of tables</b>	<b>29</b>



# 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 conveying 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, cylinders 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-dimensional 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 transferred 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 application 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 capabilities. 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 nonfunctional 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 system 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]

- problem analysis
- state of the art, problem statement
- literature research (all sources in the thesis have to be referenced [2, 1, 3, 4])
- description of existing solutions (also scientific ones, if the problem is scientifically researched), algorithms, location of the thesis in the scientific domain

Mathematical formulae

$$y = \frac{\partial x}{\partial t} \tag{2.1}$$

and single math symbols  $x$  and  $y$  are typeset in the mathematical mode.

body of the definitions



# 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 task. 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 used 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

- hardware and software requirements
- installation procedure
- activation procedure
- types of users
- user manual
- system administration
- security issues
- example of usage
- working scenarios (with screenshots or output files)

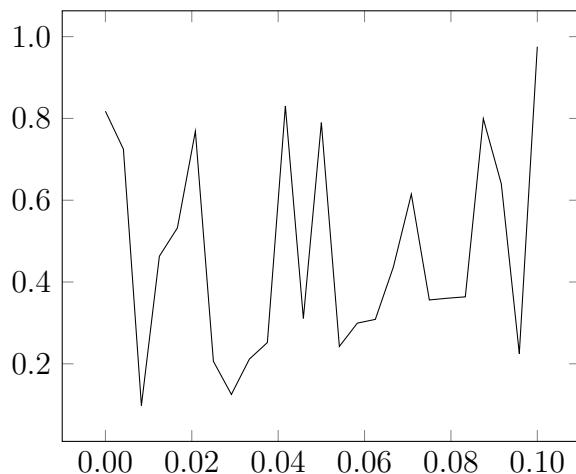


Figure 4.1: Figure caption (below the figure).



# 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.

---

```
1 class test : public basic
2 {
3     public:
4         test (int a);
5         friend std::ostream operator<<(std::ostream & s,
6                                         const test & t);
7     protected:
8         int _a;
9
10 };
```

---

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.

$\zeta$	method							
			alg. 3			alg. 4, $\gamma = 2$		
	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] Name Surname and Name Surname. *Title of a book*. Hong Kong: Publisher, 2017. ISBN: 83-204-3229-9-434.
- [2] Name Surname and Name Surname. ‘Title of an article in a journal’. In: *Journal Title* 157.8 (2016), pp. 1092–1113.
- [3] Name Surname, Name Surname and N. Surname. ‘Title of a conference article’. In: *Conference title*. 2006, pp. 5346–5349.
- [4] Name Surname, Name Surname and N. Surname. *Title of a web page*. 2021. URL: <http://somewhere/on/the/internet.html> (visited on 30/09/2021).



# **Appendices**



# Index of abbreviations and symbols

DNA deoxyribonucleic acid

MVC model–view–controller

$N$  cardinality of data set

$\mu$  membership function of a fuzzy set

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

$\mathcal{L}$  Laplace transformation



# Listings

(Put long listings here.)

---

```
1 if (_nClusters < 1)
2   throw std::string ("unknown number of clusters");
3 if (_nIterations < 1 and _epsilon < 0)
4   throw std::string ("You should set a maximal number of
      iteration or minimal difference --- epsilon .");
5 if (_nIterations > 0 and _epsilon > 0)
6   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

4.1	Figure caption (below the figure).	9
5.1	Pseudocode in <code>listings</code> .	12



# List of Tables

6.1 A caption of a table is <b>above</b> it. . . . .	14
------------------------------------------------------	----