Daniel Lines

Teaching Geometric Transformations in WebGL

Master's thesis in Computer Science
Supervisor: Theoharis Theoharis
June 2021

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Daniel Lines

Teaching Geometric Transformations in WebGL

Master's thesis in Computer Science
Supervisor: Theoharis Theoharis
June 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Acknowledgements

I would like to extend my thanks to people who have played a part in helping me complete this project. Thanks to my supervisor Theoharis Theoharis for supplying the idea for this thesis and for his guidance throughout the project. Additionally, I would like to thank my employer APEX IT AS for a great learning experience that has helped considerably with the practical part of this project.

Abstract

Geometric transformations are prevalent across numerous subjects. Understanding geometric transformations is a matter of learning how the underlying mathematical concepts are applied to obtain certain results. Previous work related to teaching these concepts have been focused on how geometric transformations are applied to computer graphics.

This work focused on defining a teaching model derived from earlier results from the field of educational technologies. The main contribution of this work is a prototype based on the derived teaching model, ideas from earlier work, and additional theory from multimedia teaching. Additionally, propositions for future research using the prototype has been made.

Sammendrag

Geometriske transformasjoner finnes i en rekke felt. Man lærer å forstå disse transformasjonene ved å lære de underliggende matematiske konseptene, og hvordan de brukes for å oppnå visse resultater. Tidligere arbeider relatert til å lære bort dette emnet digitalt har fokusert på hvordan disse transformasjonene blir brukt i visuell databehandling.

Dette prosjektet har fokusert på å definere en læringsmodell basert på resultater fra pedagogiske teknologier. Hovedbidraget er en prototype basert på den deriverte læringsmodellen, ideer fra tidligere arbeid, og annen teori fra feltet multimediaundervisning. I tillegg har forslag til fremtidig forskning med prototypen blitt fremmet.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 1 contains an introduction to the thesis.

1.1 **Background**: A short description of the background.

1.2 **Goals**: An overview of the goals for the project.

1.3 **Thesis**: Brief presentation of the structure of the thesis.

1.1 Background

Geometric transformation is a subject that has been found particularly hard to teach and learn. In part, this is due to the result of a transformation being hidden behind numerous calculations. Performing this by hand is error-prone and the result is not easily visualized. This makes internalizing these concepts a lengthy process for the students.

1.1.1 Specialization Project

The specialization project focused on researching earlier attempts at solutions for this problem. In addition, this project included investigations into how educational technologies could be incorporated to create a teaching application with an enhanced aspect of pedagogy. The result of the specialization project was a prototype showcasing the approach for geometric transformations in 2 dimensions.

1.2 Goals

The main goal of this thesis is to build on the work from the specialization project. Specifically, developing a prototype which uses the teaching model to teach 3D geometric transformations in an intuitive way. The prototype should be developed for the web as it is easily accessible and cross-platform.

In addition to the main goal, a suitable mapping of the teaching model outlined in the specialization project should be made. This is required to ensure the efficiency of the teaching model.

Research into the viability of the teaching model is beyond the scope of the thesis. Regardless, suggestions for future investigations will be given.

1.3 Thesis

This thesis has the following structure. Chapter 2 describes the theoretical background for the work. A description of the methodology applied is given in Chapter 3. Chapter 4 presents the resulting prototype of the development work. An evaluation of the prototype and the methodology is given in Chapter 5. The possibilities for future work is outlined in Chapter 6. Chapter 7 discusses the conclusions reached.

Chapter 2

Theory

This chapter gives an overview over the theory used in the thesis.

2.1 **Framework**: Describes the digital educational framework the application is based on.

2.2 **Cognitive Multimedia Theory**: Presents the cognitive theory of multimedia learning.

2.3 **Gamification**: Briefly describes the term gamification.

2.4 **Mathematical Theory**: Gives a brief overview of mathematical theory applied in the development.

2.5 **Related Work**: Reports on related work.

2.6 **Material Design** Introduces the design framework Material Design.

2.7 **Technologies**: Overview of relevant technologies.

2.8 **Security**: Describes relevant concepts from security.

2.1 Framework

In the specialization project it was decided to use a framework as a guide for the design of the application in regard to user interaction. This was done to ensure it will be effective in regard to teaching, and grounded in theory from the field of educational technologies. The chosen framework was the one proposed by Leftheriotis. et. al. which was aimed to teach students about plant biology [3]. The results of the study performed using the framework were positive with respect to engagement, motivation and learning outcomes [3].

2.1.1 Structure of the framework

The exploration stage of the framework describes a process where the student familiarizes her/himself with the subject by looking through annotated pictures of plants [3]. An application stage describes answering questions drawn from the subject the student was studying in the previous stage [3]. Lastly, the creation stage is described as reproducing annotated drawings of plants from memory [3]. A brief description of the framework can be seen in Table 2.1.

Table 2.1: Overview of the framework [3]

Stage	**Description**
Exploration	Student familiarizes with artifacts based on the subject
Application	Students uses the knowledge to answer questions
Creation	Students recreates artifacts from memory

These stages were based in the three epistemological traditions, *Behaviorism*, *Cognitivism* and *Constructivism* [3]. A short definition of these traditions is shown in Table 2.2.

Table 2.2: Definition of epistemological traditions [3]

Tradition	**Definition**
Behaviorism	Gaining knowledge through experience
Cognitivism	Gaining knowledge through experience and thinking
Constructivism	Constructed knowledge

2.2 Cognitive Multimedia Theory

Teaching subjects through digital media requires presenting information in an optimized way for learning. According to Mayer there are three base assumptions for the cognitive theory of multimedia learning [4]. (1) Humans process auditory and visual information through two separate channels, (2) there exists a limitation to the amount of information that can be assimilated concurrently through these channels, (3) active learning consists of combining mental representations of relevant incoming information with existing knowledge [4]. The theory shown in Figure 2.1 is based on these assumptions.

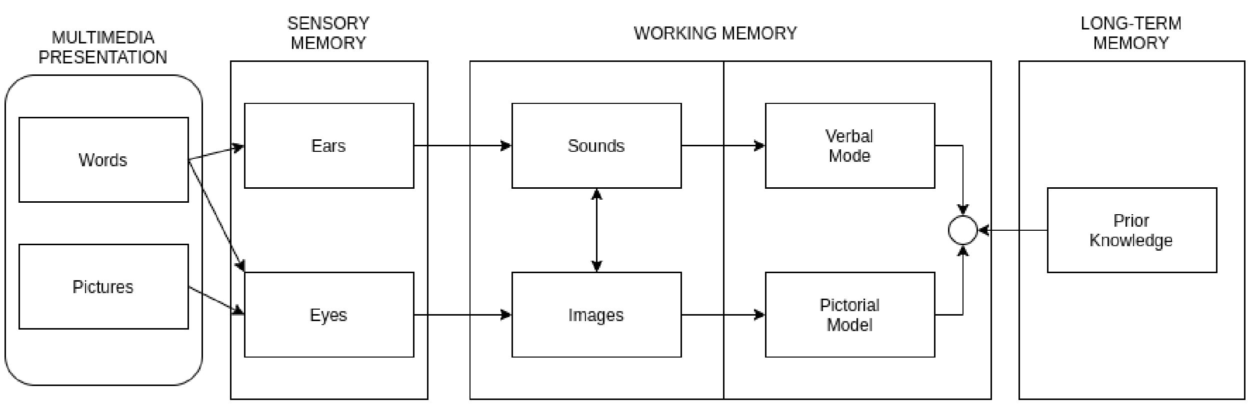

Figure 2.1: Cognitive theory of multimedia learning [4]

This model suggests using pictorial information while minimizing textual representations will optimize information absorption [4].

2.3 Gamification

The term *gamification* refers to including elements from game design into an application [5]. It has been shown that gamification has had an advantageous effect on cognitive learning outcomes, engagement and motivation [6]. Leftheriotis et al. also found positive results using gamification techniques [3].

2.4 Mathematical Theory

Mainly, this thesis relies on mathematical theory from linear algebra, with a focus on transformations as applied in computer graphics. This section gives a brief overview of these theoretical concepts. See Appendix B for more detailed information about the specific matrix representation of the transformations used.

2.4.1 Affine Combinations

An affine combination is a point $p \in \mathbb{R}^3$ such that,

$$p = \sum_{i=0}^{n} a_i p_i \tag{2.1}$$

where $a_i \in \mathbb{R}$ and $p_i \in \mathbb{R}^3$ for $i = 0, 1, ..., n$ [7]. An additional constraint is that $\sum_{i=0}^{n} a_i = 1$ [7].

A *convex polygon* is an example of an affine combination which is used as the building block of 3D models (meshes) [7].

2.4.2 Affine Transformations

An *affine transformation* is a transformation that preserves affine combinations [7]. Thus, a transformation T is affine if,

$$T(p) = \sum_{i=0}^{n} a_i T(p_i) \tag{2.2}$$

2.4.3 Homogenous Coordinates

Homogenous coordinates uses on more dimension than the target space [7]. This homogenizes the transformation matrices which means we can represent the relevant 3D transformations as 4 matrices [7].

2.5 Related Works

This section gives a quick overview of some earlier efforts into making an interactive tool for teaching geometric transformations. These particular works were selected based on their approach and implemented ideas.

2.5.1 I3T

I3T is an interactive tool for studying 3D geometric transformations [1]. By incorporating a graph tool, the user can add transformations to 3D objects, their camera, and perspective [1].

As seen in Figure 2.2, there is also a visualization of the projection plane, and the view frustum. Earlier research on using I3T for teaching computer graphics found the learning curve of I3T to be factor in its efficiency [8].

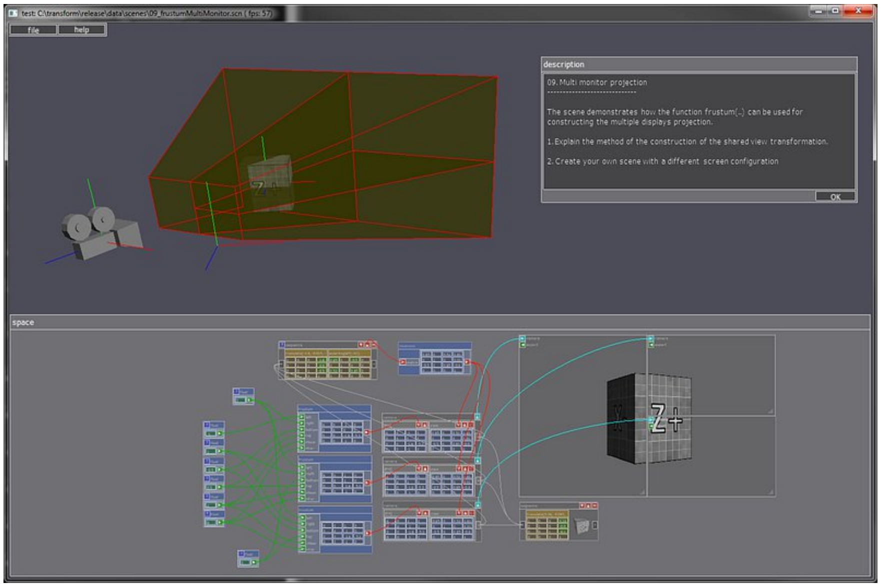

Figure 2.2: A view from the I3T tool. [1]

2.5.2 AR Tool

Thomas Suselo et al. developed a tool for teaching 3D geometric transformations in AR [2]. By using AR the student could see the 3D object they were adding transformations to in a familiar space, and after adding transformations the OpenGL code would be displayed [2]. In addition, small tasks were added were the student could add transformations to match a 3D objects transformations to a target object [2].

Figure 2.3 shows the AR tools screen.

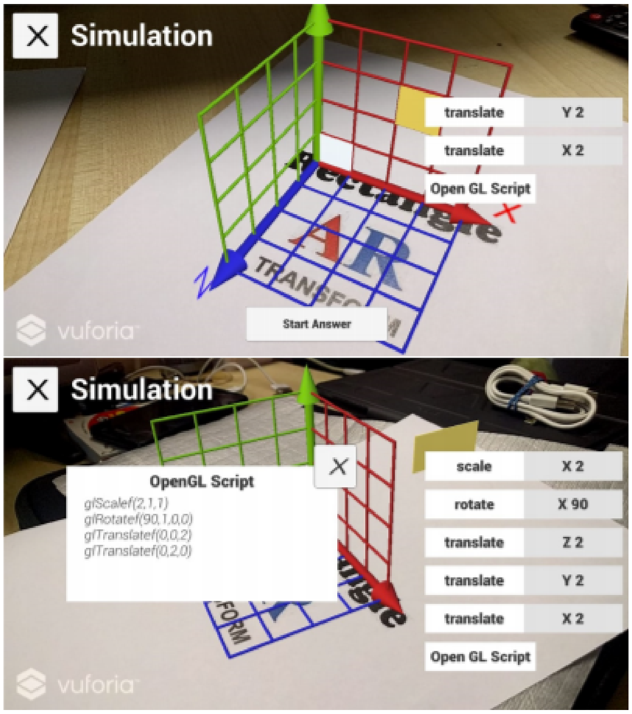

Figure 2.3: AR tool for teaching geometric transformations [2]

2.6 Material Design

Google has developed *Material Design* as a framework for creating application with high-quality design [9]. Using a design framework consistent with industry guidelines is highly advantageous for usability. In addition, it creates a familiar look and feel straight away for new users. Material design is aimed at different platforms [9], which creates a uniform look regardless of which platform the user is accessing the product on.

2.7 Technologies

2.7.1 TypeScript

JavaScript is a widely used programming language on several platforms [10]. TypeScript builds further on JavaScript to create a strictly typed programming language that can be compiled down to JavaScript [11]. To ensure maintainability and stability, TypeScript was chosen as the main programming language for the project.

2.7.2 Frameworks

Utilizing frameworks can speed up the development process, while maintaining quality and performance of the finished product. What follows is an overview of the frameworks utilized in the development of the application.

2.7.2.1 Marble.js

Marble.js is a backend framework written for Node.js with reactivity in mind as well as functional design principles [12]. It was chosen for its functional design and because it is written in TypeScript.

2.7.2.2 Angular

Angular is a powerful frontend JavaScript framework with a rich set of tools and features [13]. It was chosen because it had all the features meeting the requirements of the application. Angular supports deploying the application on different platforms (web, mobile, desktop)[13], which creates great flexibility for future expansion of the application.

2.7.3 WebGL

WebGL is a low-level graphics API for the web browser based on OpenGL ES [14]. This allows for interacting with the GPU from JavaScript, developing interactive 3D applications that run in a web browser.

2.7.4 MongoDB

MongoDB is widely used NoSQL database [15]. It was chosen as data persistence because it is highly scalable and easily integrated with the rest of the technologies used.

2.8 Security

Security is an essential part of any system that requires processing sensitive data such as personal data. What follows is a short description of relevant security techniques for this thesis.

2.8.1 JSON Web Tokens

JWT represents a secure way of transmitting information between two parties [16]. They are widely used for *authorization*, meaning a user can only access certain resources if a valid token is provided [16].

2.8.2 GDPR

The General Data Protection Regulation (GDPR) is a regulation meant to protect users from misuse of personal data [17]. This regulation pertains to any entity that processes user data from the EU, regardless of the location of said entity [17].

Chapter 3

Methodology

Chapter 3 is a description of the methods used in the development of the application.

3.1 **Teaching Model**: Describes how the educational framework is mapped on the domain of geometric transformations.

3.2 **Constraints**: Short description of constraints.

3.3 **Design**: Design methodology.

3.4 **Requirements** Overview of the requirements.

3.5 **Architecture**: Describes the architecture of the proposed prototype.

3.6 **Solving Problems**: Presents choices for implementing problems for students to solve.

3.1 Teaching Model

It was determined in the specialization project that the teaching model for the application should be derived from the framework proposed by Leftheriotis et al. This meant defining three activities, (1) *exploration*, (2) *application* and (3) *creation* mapped from the framework

Table 3.1: Overview of the activities in the application

Stage	**Original**	**Mapping**
Exploration	Students watched annotated pictures of plants biology	Students adds transformations to a 3D object and can manipulate only relevant values based on type of transformation (e.g. θ in a rotation, x, y, z in translation/scale)
Application	Students answered questions based on the subject	Students add transformation to a 3D object to match the transformation of a second 3D object
Creation	Students drew artifacts based assisted by connect-the-dots guidelines	Students can add transformations to a 3D object and be able to freely manipulate values in the matrix

stages. The framework was originally tested with the domain of plant biology [3], which is fundamentally different from the target domain of geometric transformations. Thus, special care was needed to ensure the efficiency of the framework was not lost in the mapping. A transformation matrix was chosen as the artifact most analogous to the annotated pictures used in the original paper. Table 3.1 shows an overview of the selected mapping.

3.1.1 Exploration

In the original paper the exploration stage had students viewing annotated pictures of plants to give an introduction to the subject [3]. Matrices representing the transformations should in this stage be presented to the student such as the annotated pictures of plant biology were presented in the original paper. What sets the matrices apart and determines which transformation they represent, are which values that can change without altering what kind of transformation they will apply to the target point. Thus, these values could perform the same function for their respective matrices, as the annotations do for their pictures.

3.1.2 Application

The application stage consisted of a simple quiz game for increasing the students' understanding of the artifact [3]. Using a similar approach as the AR tool developed by Suselo et al. [2], the quiz is replaced by a problem where the objective is to recognize which composite transformation has been added to a target 3D object. The student solves this problem by building this composite transformation out of the transformations that were explored in the preceding stage. In addition to the target object, an object the student is adding transformations to should also be visualized.

3.1.3 Creation

In the *creation* stage the students drew artifacts assisted by connect-the-dots guidelines [3]. For matrices, a suitable replacement is allowing student to manipulate freely chosen values in the matrix and experience the effect visually. With the experience from the previous stages, the student should be able to recreate transformation matrices further cementing these into memory.

3.2 Constraints

It is preferable constraining the subject to a number of transformations. The transformations used are,

- Scaling
- Translation
- Rotation around axis (x, y, z)
- Shearing on axis (x, y, z)

More on the matrices for these respective transformations in Appendix B

3.3 Design

Textual representations should be minimized favoring pictorial representation where possible. This is based on the *CTML* presented in Chapter 2 and should optimize the students' ability to absorb and understand the concepts. Additionally, using *Material Design* should minimize confusion related to interaction with the UI.

3.4 Requirements

A set of functional and technical requirements for the application were drafted. The functional requirements are outlined in Tables 3.2 and 3.3. The technical requirements are outlined in Table 3.4.

These requirements were based in the educational framework mapping from Section 3.1.

Table 3.2: General Functional Requirements

ID	Requirement	Priority
FR1.1	The user can log into the application	HIGH
FR1.2	The user can log out of the application	HIGH
FR1.3	The user can view the data generated by the application	MEDIUM
FR1.4	The user can navigate between the sections of the application	HIGH
FR1.5	The application should generate click-streams	LOW
FR1.6	The application should be able to tell when a session has ended	LOW
FR1.7	The application should ask the users' permission to generate data	HIGH
FR1.8	The application should not generate data unless it has the users' permission	HIGH
FR1.9	The user can register as a user of the system	LOW
FR1.10	The user can change their password	LOW

Table 3.3: Functional Requirements for Transformations

ID	Requirement	Priority
FR2.1	The user can add transformations to a 3D object	HIGH
FR2.2	The user can add matrices to a transformation	HIGH
FR2.3	The user can remove matrices from a transformation	HIGH
FR2.4	The user can manipulate the relevant values of a matrix	HIGH
FR2.5	The user can view the projection of a transformed 3D object unto a projection plane	LOW
FR2.6	The user can manipulate relevant values of the projection	LOW
FR2.7	The user can see the original orientation of a 3D objects' faces	HIGH
FR2.8	The user can solve problems by adding necessary transformations	HIGH
FR2.9	The user is notified when a problem is solved	HIGH
FR2.10	The user can add matrices and manipulate a chosen value	HIGH
FR2.11	The application should show the transformations in the pipeline (model, projection)	LOW
FR2.12	The application should be able to tell how much time a user has spent in a section	HIGH
FR2.13	The application should lock sections until the user meets the requirements to use them	HIGH
FR2.14	The application should be able to tell how many problems a user has solved	HIGH
FR2.15	The application should be able to visualize the coordinate axis	HIGH

Table 3.4: Technical Requirements

ID	**Requirement**	**Priority**
TR1	The application should be compatible with modern browsers	MEDIUM
TR2	The application should be developed with the chosen technologies	HIGH
TR3	The application should comply with GDPR	MEDIUM

3.5 Architecture

Keeping track of a students progress through the framework stages requires persisting this data between user sessions. A simple implementation for registering a user and logging in and out meets the requirements for a prototype.

Developing a backend API for persisting data is a good choice because et decouples the backend from the front end, ensuring better maintainability of the application in the future. The proposed architecture for the prototype is presented in Figure 3.1.

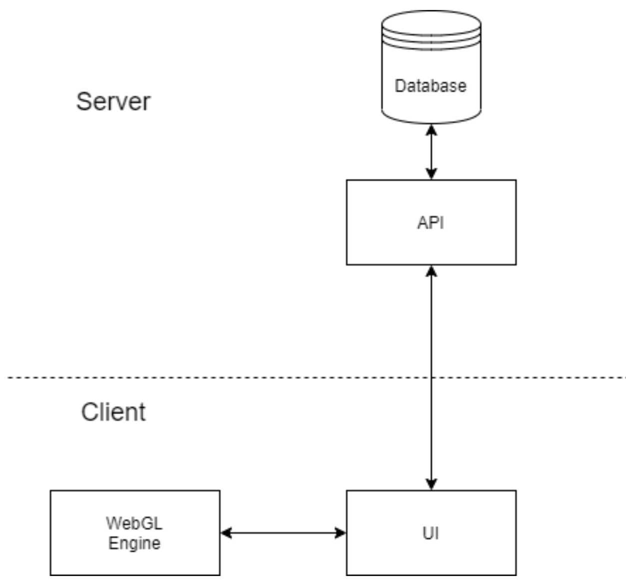

Figure 3.1: Architecture of the prototype.

3.5.1 Rendering Engine

Writing a full featured general rendering engine for WebGL is well beyond the scope of this thesis. Regardless, having generality in mind when creating the rendering engine has several advantages in regard to future expansion. For this use case, support for rendering textured meshes in a 3D scene context is sufficient. Additionally, a simple interface for manipulating the transformations of these meshes is needed. More sophisticated features (e.g. proper lighting, shadows, occlusion culling) are not necessary for the use case at this point in time. Figure 3.2 shows a simplified representation of the proposed rendering engine.

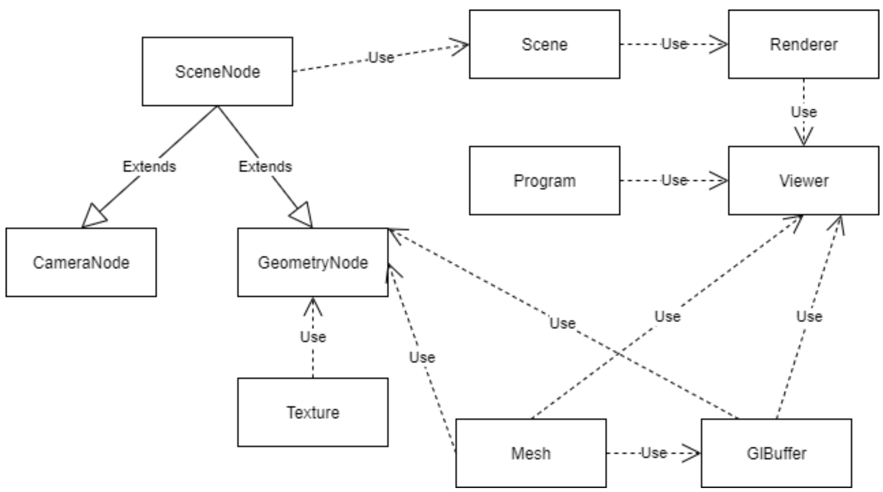

Figure 3.2: WebGL rendering engine architecture.

3.6 Solving Problems

Section 3.1.2 mentions the students are asked to solve problems in the *application* stage of the application. In this context, a problem is represented by a target state for the 3D object the student will add transformations to. This target state is visually represented as a copy of the main 3D object with a different texture.

Adding transformations to the target mesh is done by generating a random integer n. For each $i = 1, 2, ..., n$ add a random transformation to the target mesh.

3.6.1 Checking Problem State

The problem is considered solved once the 3D objects state is sufficiently close to the target state.

Ensuring the main objective of understanding the transformations, a simple cube mesh is used as the 3D objects. Testing whether the correct transformations have been added to the mesh consists of testing whether there is sufficient overlap of its state with the state of the target mesh.

Let $\boldsymbol{M} = \boldsymbol{v}_0, \boldsymbol{v}_1, ..., \boldsymbol{v}_n$ be the set of vertices in the mesh the student has control over, and $M_t = \boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_n$ be the set of vertices in the mesh representing the target state. For the proposed method, a problem is considered solved if $d = |\boldsymbol{T}(\boldsymbol{v}_i) - \boldsymbol{u}_i| < c$ for $i = 0, 1, ..., n$. T is the composite transformation added by the student and the threshold $c \in \mathbb{R}$ is a constant determining how closely the states have to overlap. It is safe to compare the two sets like this, because the target mesh and the main mesh are copies of each other, and the transformations are *affine*.

Chapter 4

Prototype

Chapter 4 presents the prototype and necessary development details.

4.1 **Introduction**: A brief introduction on the functionality provided by the prototype.

4.2 **User System**: Short description of the user system in the prototype.

4.3 **Main View**: Describes the components of the main view.

4.4 **Activities**: A more detailed description of the activities.

4.5 **Rendering Engine**: Some details on the rendering engine.

4.6 **Profile Page**: A presentation of the profile page.

4.7 **Help Dialogs**: A brief description of the help dialogs.

4.8 **Requirements**: Overview of the requirements met by the prototype.

4.1 Introduction

As described in Section 3.5, the prototype is divided into a back-end API and a front-end. The API is implemented using Marble.js [12] and uses MongoDB [15] for data perstance.

Angular was used as a framework for the front-end.

4.2 User System

Requirements presented in Section 3.4 meant the need for persisting the students' progress over time. This required developing a simple system for logging into the application before using it.

The prototype has simple functionality for registering a user, and logging into the application. A user is uniquely associated with the email they provided upon registration.

4.2.1 Security

Securing the user system is done by authorizing every request made to the API using JSON Web Tokens (JWT). After the user is authenticated when providing their email and password, the front end receives a security token which has to be provided for all communication with the API. JWT are widely used as a method for securing API's in this way [16].

Additional security measures include,

- A password strength requirement upon registration
- Password are encrypted before they are stored in the database.
- Automatic logout from frontend upon receiving an Unauthorized error response from the API.

4.3 Main View

The main view of the prototype consists of a menu far to the left for navigation. A main activity window has a 3D view to the left and a control panel to the right. For performance issues, the activities that represent the stages in the framework are gathered under the same view. This ensures the prototype can use the same WebGL context while the student is using the activities. The WebGL documentation contains more information on how WebGL contexts work [18].

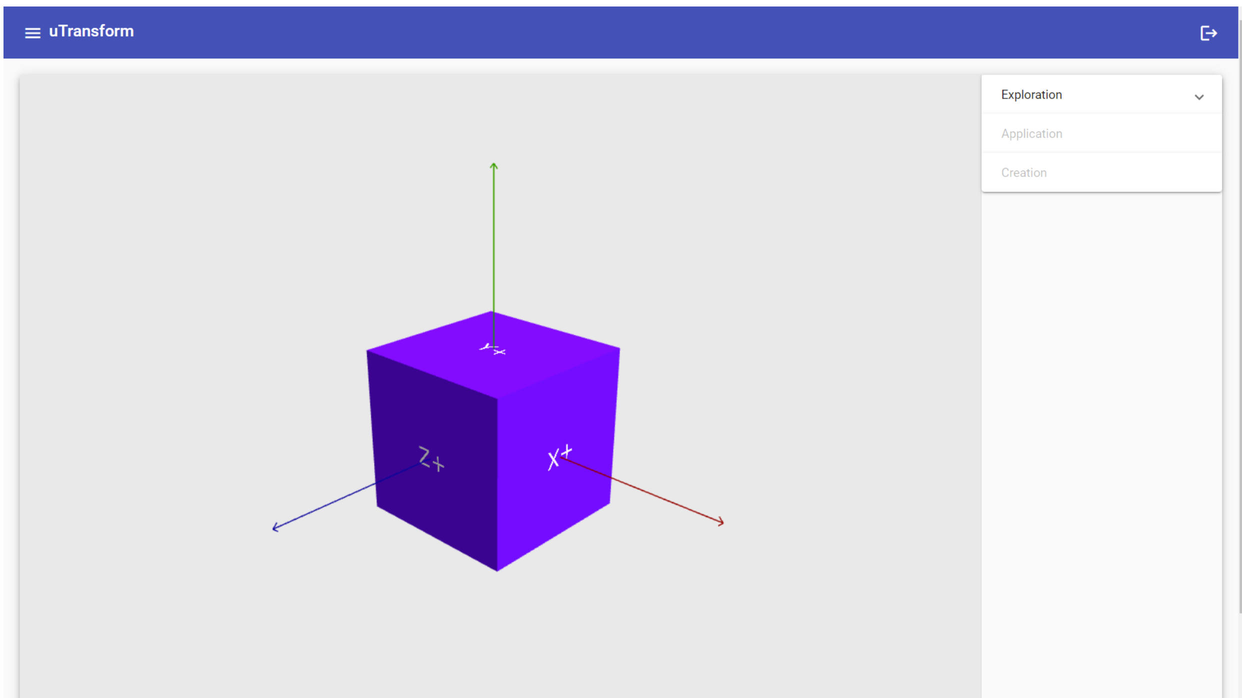

Figure 4.1: Main view of the prototype

In the 3D view window, there are additional camera-controls for pivoting around origo, and zooming. Coordinate axis and textures marking the original orientation of the 3D object is added to ensure the user can move the camera without losing track of the orientation. Figure 4.1 presents the main view of the prototype.

4.4 Activities

There are three activities corresponding to the three stages of the framework. From an implementation point of view, these activities differ in how the control panel works, and how the application reacts to user actions. Initially, only the first activity is available until the user makes the necessary progression. This is visually indicated in Figure 4.1 by the choices being disabled.

4.4.1 Exploration

For the exploration activity, the control panel consists of a list of letters representing the transformation matrices. Figure 4.3 shows the exploration activity. A matrix is displayed once its corresponding letter has been clicked, which also brings up a set of sliders corresponding to the relevant values that can be manipulated. Adding another transformation requires the user to choose type of transformation, the position of the transformation in the list relative to the selected transformation, and in some cases additional information such as the axis of a rotation as shown in Figure 4.2 This info is supplied through a dialog that opens up when a user clicks on the plus symbol left of the transformation list. Selecting a transformation allows the user to manipulate the relevant values, and remove the transformation from the list. Figure 4.4 gives a closer look at the control panel for exploration.

These transformations correspond to the model transformation, which is indicated in the transformations list by equating them with a composite transformation M.

The prototype persists how much time has been spent in the exploration activity. Idle time is taken into account to prevent users from progressing through the stages without spending actual time interacting with the system. This safe-guard is elementary, and only takes into account that the user is not idle.

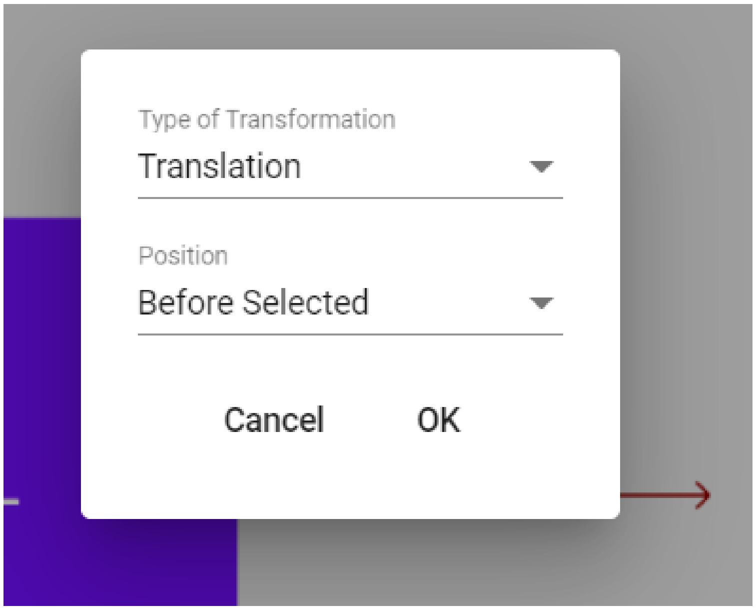

Figure 4.2: Transformation add dialog

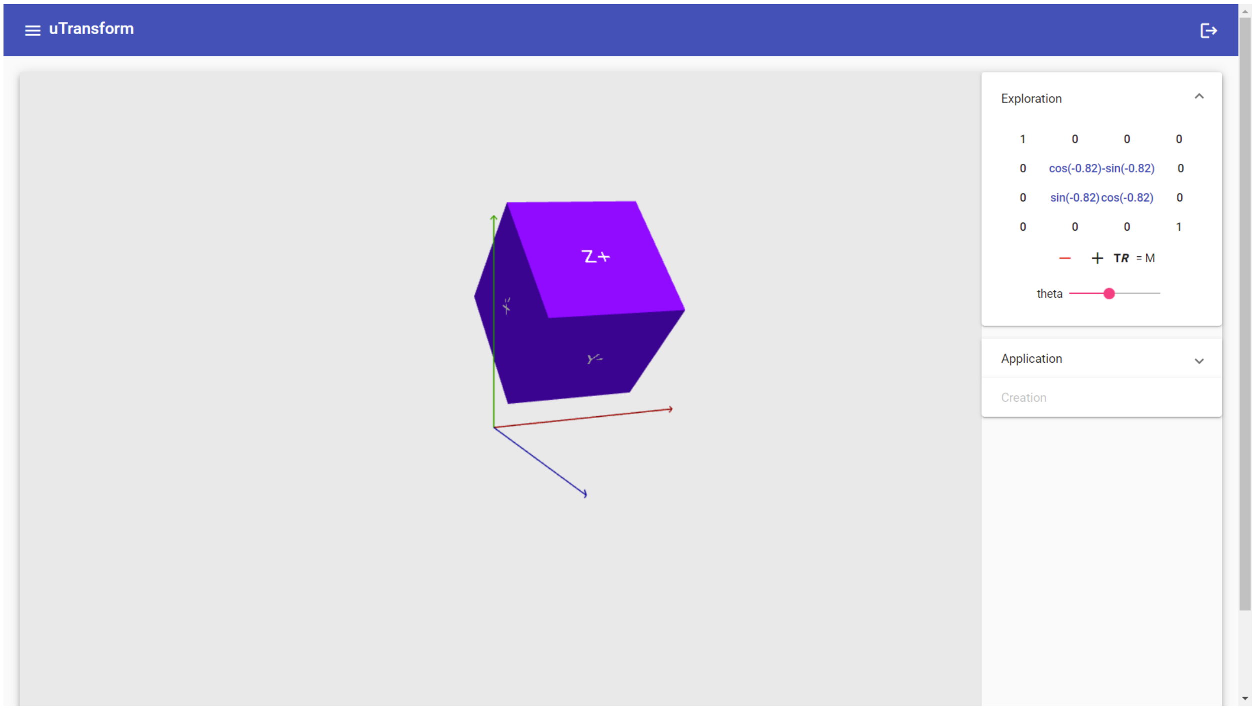

Figure 4.3: Exploration activity

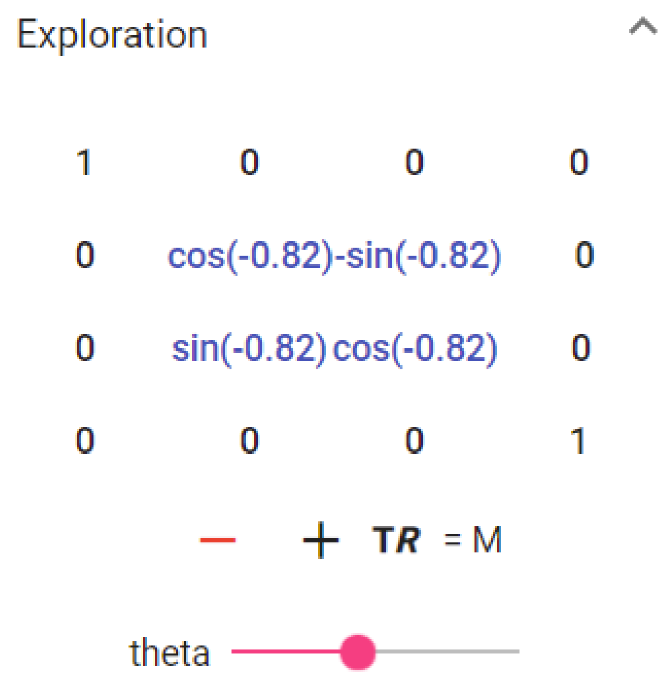

Figure 4.4: Control panel for exploration

4.4.2 Application

The application activity becomes available to a user once they have spent the required amount of time in the exploration activity. It shares the same control behavior as the exploration activity, but adds a copy of the main 3D object with a different texture. This is referred to as the target object, and has a series of random transformations automatically applied to it. Figure 4.5 shows the two objects together in the scene. Adding a set of transformations which transforms the main 3D object in the same way as the target object, will notify the user that the problem has been solved, and restart with another set of target transformations. The prototype persists a counter over how many problems have been stored.

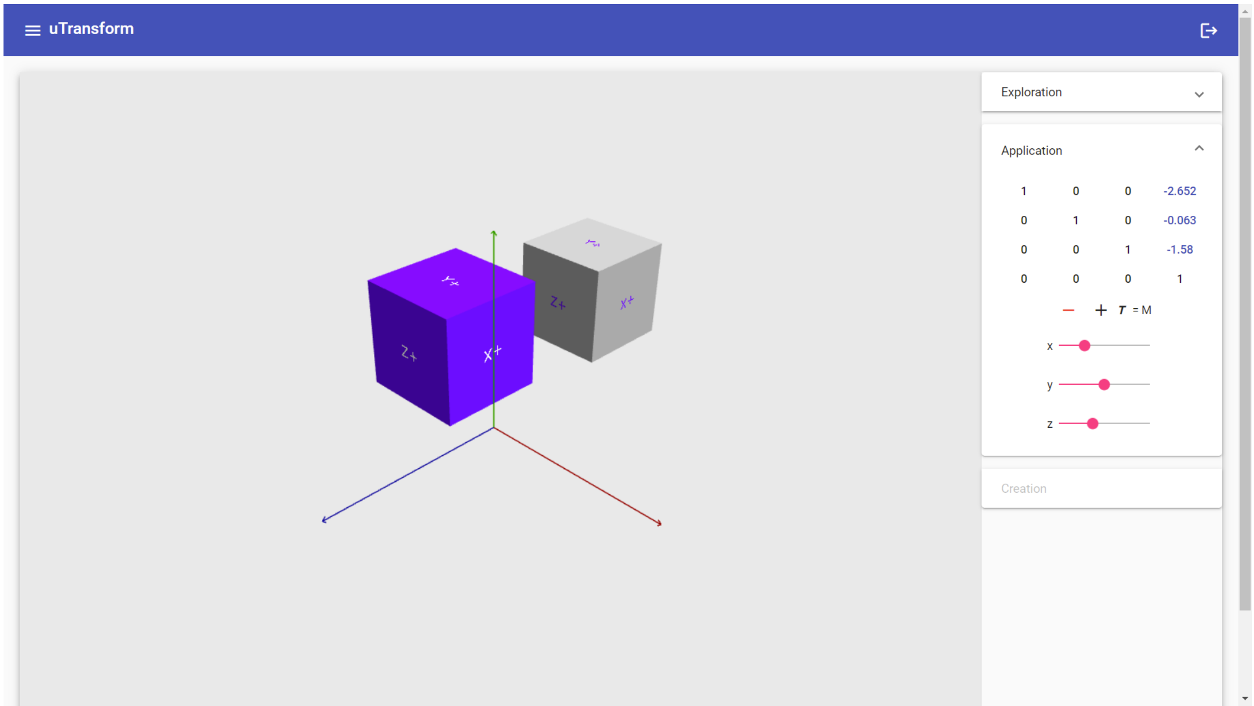

Figure 4.5: Application activity

4.4.3 Creation

In the creation activity, the control panel behavior has been modified such that the user can choose which value in the matrix to manipulate. Once a value has been clicked a slider will appear for the chosen value. This is illustrated in Figure 4.6.

The values in the bottom row of the matrix can not be chosen because they are a result of using homogenous coordinates. Allowing the user to manipulate these values could be a source of confusion.

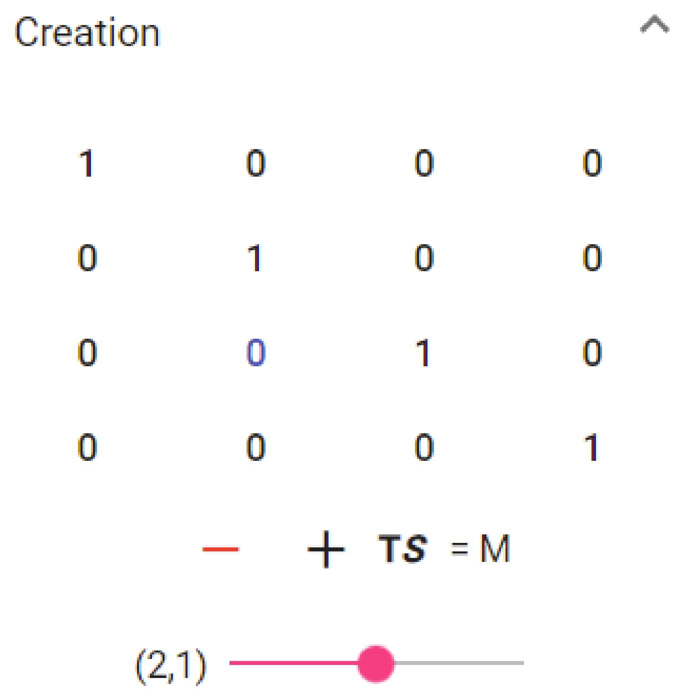

Figure 4.6: Control panel for creation activity

4.5 Rendering Engine

A rendering engine was developed for rendering simple objects in 3D, its architecture presented in Section 3.5. The main view interfaces with the rendering engine by sending the necessary transformations based on user interaction. By sending new values only in the event they are changed by the user, some calls to the GPU are saved.

Effort has been made to create a decoupling between the rendering engine and the user interface. Future additions to the application will benefit from this, specifically when they are expansive (i.e. expanding the functionality beyond the original vision).

4.6 Profile Page

A simple profile page has been added as a way of promoting data transparency. The main function of this page is to present the user data that has been stored pertaining to the user that is logged in. It contains a simple key-value list describing the data as seen in Figure 4.7.

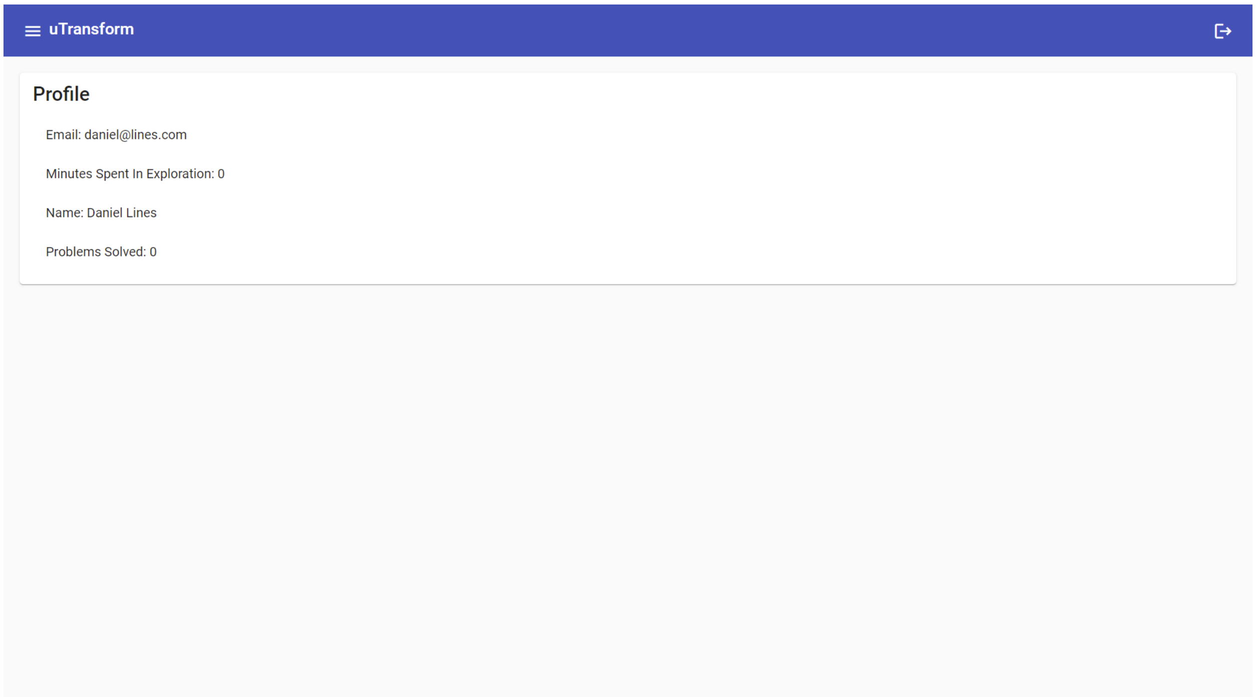

Figure 4.7: Profile page view

4.7 Help Dialogs

Each of the activities have a button which will display a dialog informing the user of the functionality. Additionally, it contains some helpful tips on how to interact with the matrices. This is a measure for improving the usability of the prototype, and to avoid frustrations stemming from users misinterpreting how a certain functionality is supposed to work. Figure 4.8 shows one of these help dialogs

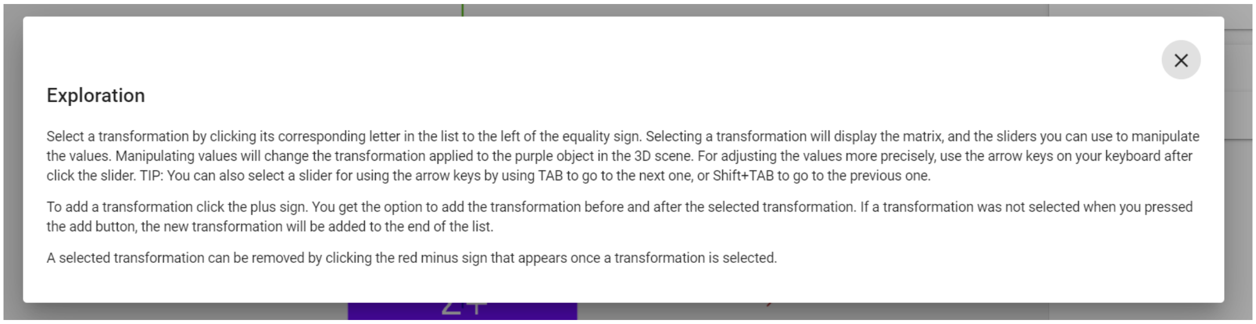

Figure 4.8: Example help dialog

4.8 Requirements

A set of functional and technical requirements were presented in Chapter 3. This represented the original vision for the functionality. The prototype does not meet all of these requirements. Tables 4.1 and 4.2 show the revised set of requirements that are met by the prototype. Requirements which are not met by the prototype are marked in red. Chapters 5 and 6 touches on reasons certain features were left out of the prototype.

Table 4.1: Revised general functional requirements

ID	**Requirement**	**Priority**
FR1.1	The user can log into the application	HIGH
FR1.2	The user can log out of the application	HIGH
FR1.3	The user can view the data generated by the application	MEDIUM
FR1.4	The user can navigate between the sections of the application	HIGH
FR1.5	The application should generate click-streams	LOW
FR1.6	The application should be able to tell when a session has ended	LOW
FR1.7	The application should ask the users' permission to generate data	HIGH
FR1.8	The application should not generate data unless it has the users' permission	HIGH
FR1.9	The user can register as a user of the system	LOW
FR1.10	The user can change their password	LOW

Table 4.2: Revised functional requirements for transformations

ID	**Requirement**	**Priority**
FR2.1	The user can add transformations to a 3D object	HIGH
FR2.2	The user can add matrices to a transformation	HIGH
FR2.3	The user can remove matrices from a transformation	HIGH
FR2.4	The user can manipulate the relevant values of a matrix	HIGH
FR2.5	The user can view the projection of a transformed 3D object unto a projection plane	LOW
FR2.6	The user can manipulate relevant values of the projection	LOW
FR2.7	The user can see the original orientation of a 3D objects' faces	HIGH
FR2.8	The user can solve problems by adding necessary transformations	HIGH
FR2.9	The user is notified when a problem is solved	HIGH
FR2.10	The user can add matrices and manipulate a chosen value	HIGH
FR2.11	The application should show the transformations in the pipeline (model, projection)	LOW
FR2.12	The application should be able to tell how much time a user has spent in a section	HIGH
FR2.13	The application should lock sections until the user meets the requirements to use them	HIGH
FR2.14	The application should be able to tell how many problems a user has solved	HIGH
FR2.15	The application should be able to visualize the coordinate axis	HIGH

Chapter 5

Evaluation

Chapter 5 contains a discussion around the proposed methodology and the resulting prototype of the development.

5.1 **Teaching Model**: Discussion surrounding the derived teaching model.

5.2 **GDPR**: Arguments around the need for GDPR compliance.

5.3 **User Testing**: Touches on problems around user testing, and why user tests were not implemented.

5.4 **Implementation**: Evaluation of the implementation of the teaching model.

5.5 **Data Collection**: Brief discussion of the data collection method.

5.6 **User System**: An evaluation of the user system.

5.7 **Design**: A short discussion on the approach on the design of the prototype.

5.1 Teaching Model

There are numerous ways the educational framework developed by Leftheriotis et al. could have been mapped to the domain of geometric transformations. The reasoning behind the selected mapping focused on retaining the function of the stages, as opposed to using a one-to-one mapping. The third assumption of CTML [4] implies that humans absorb knowledge in a piece-wise fashion, integrating newly acquired knowledge with earlier accumulated knowledge. Thus, separated activities (stages) that contain a finite amount of new knowledge will facilitate this accumulation of knowledge.

An analysis of the main function of each stage in the framework based on the paper by Leftheriotis et al. [3] is shown in Table 5.1.

Table 5.1: Main function of each stage of the framework

Stage	**Function**
Exploration	Exploring a subset of the subject with constraint based on the teaching goal
Application	Using the experience obtained from the Exploration stage for problem-solving
Creation	Recreating artifacts from the first stage

The *exploration* stage is centered around familiarizing the student with the subject. A reasonable approach to this within geometric transformations is to let the student explore how the different transformations affect a 3D object. By constraining manipulation of matrix values only to those relevant for each type of transformation, the student will be familiarized with which matrix values are relevant. In addition, the student will gain experience in the effect each type of transformation has on a 3D object.

For the *application* stage, the main function is to apply the experience and knowledge obtained in the previous stage. Using the same constraints as the *exploration* stage will let the student focus on obtaining problem-solving experience. Refraining from introducing new constraints work to avoid overwhelming the student.

In the *creation* stage, the constraints on which values to manipulate are not present. The main function of this stage is to recreate transformations based on experience from the previous stages. Visual confirmation of how the manipulation affects the 3D object will further cement experience and knowledge from the earlier stages.

To avoid confusion stemming from skipping too far into the process, the stages were locked behind certain achievements. The application stage would only be accessible after spending a set amount of time in the explorations stage, while the creation stage would be locked behind completing a set number of problems in the application stage. In addition to softening the learning curve, this approach has advantages bringing in an aspect of gamification. As outlined in Chapter 2, reaching goals has a positive effect on mastering a subject.

The original vision for the prototype focused more on how geometric transformations are applied in computer graphics. This is demonstrated in earlier sketches for the prototype made during the specialization project, and the drafted requirements presented in Chapter 3. These sketches are available in Appendix C. This approach was abandoned in favor of a design more focused on the derived teaching model.

5.2 GDPR

Before developing the application, a set of requirements were drafted. These are presented in Section 3.4. Some of these requirements mention complying to GDPR guidelines.

While complying to GDPR guidelines is essential for a release version, it was decided not to prioritize this during the development. The developed application is a prototype, and complying to all GDPR guidelines was not deemed to be a part of an MVP.

5.3 User Testing

The specialization project proposed performing user tests to test the viability of the prototype. While user tests are a powerful tool to ensure usability of a product, it was decided not to perform them at this stage in the development. There are several aspects of the prototype that are hard to test, because they are dependent on time, requiring the correct context, and having the correct target group. User testing could have been performed to test general usability of the UI. However, incorporating industry standards through material design should fulfill the minimum usability level required for this prototype. The following section touch on parameters of the application that are in need of testing.

5.3.1 Activity Constraints

The application and creation activities are locked behind progression. This progression is based on time spent in the exploration stage and problems solved in the application stage respectively. These parameters have not been optimized as it is beyond the scope of this thesis. Observing students as they use the prototype over time should create sufficient data for this. Additionally, whether the previous activities should be left open after the next on is unlocked should be determined by additional research.

5.4 Implementation

The prototype used the stages sequentially but allows the student to return to a previous activity at any time. This might prevent the efficiency of the application and creation activity, as the student is able to return to earlier stages for hints rather than spending time thinking about the issue. Using a fully sequential system where only one activity is open at any time in the progression is an option. However, answering which alternative yields the best result would require additional research.

5.4.1 Application Stage

An alternative to the proposed method for checking the problem state outlined in Section 3.6 would be to check if the specific transformations added by the student match those added to the target mesh. This approach would be problematic because several combinations of transformations can yield the same result. Visually, this could result in the two meshes overlapping on screen, but as the states were not achieved by the same combination of transformations, the application would not consider the problem solved.

There are limitations to the problems generated for the students to solve in this stage. They are simple, match the target problems that are easily solved once the student understands the transformations at a rudimentary level. Additionally, the problems don't require several steps to solve. A different approach, where problems are more complex, and have more steps, might ensure a deeper understanding on the subject. Having more complex problems would require more complex generation code, or hand-crafted problem templates. This more complex approach was deemed not necessary for an MVP. Future iterations of the application should implement a variation of problems for the student to solve.

Leftheriotis et al. included interactions between students at this stage with a collaborative mode, and a competitive mode [3]. Leaving these elements out of the implementation here, might cause a difference in results. However, students are free to collaborate when solving problems by applying the application while sitting in study groups, which should create a sufficiently similar approach. Further work could expand on the application activity, adding more gamifying elements as was done by Leftheriotis et al.

5.4.2 Creation Stage

When the framework was tested, the students had some guidelines during the final stage [3]. It could be argued that these guidelines are missing from the implementation for geometric transformations. However, as the students are given the format of the matrix, further guidelines (e.g. tips for which values in the matrix correspond to creating a translation matrix) might work counter-productive by limiting experimentation with different combinations. Additionally, the student is free to return to the previous stages to review the shape of the matrices if needed.

Values are manipulated through using a slider in the interface. This works well for matrices where the values are not mathematical functions (e.g. *cos(x)*). As the implementation is now, creating rotation matrices is hard. Simple manipulation of numerical values the right position is not sufficient. The values have to be constrained in the range and proportions of the correct trigonometric functions (Read more about the specific matrices used for the transformations in Appendix). A better solution would be functionality for defining functional output in the matrix slots, and then adjusting the input to these definitions.

5.5 Data collection

The specialization project [19] proposed using click-streams for collecting data. During the development stage of the prototype it was clear that using this technique was not required for the wanted functionality. Time spent in the stages and solved problems were the parameters needed to be recorded. Persisting these values works as a better solution. Additionally, this method is not as compromising to the users' privacy.

5.6 User system

The user system in the prototype represents the most essential features for a web application. Several features that are normally included into such application is left out. This is considered sufficient for the scope of this thesis, specifically because it focuses on the teaching model and its implementation. Registration might also be considered a superfluous feature in this regard. It was added for easier addition of users in a potential study.

5.7 Design

Peter Felkel et al. found the learning curve of I3T with respect to its interface to be a factor [8]. Additionally, humans have a certain limitation to their ability to process visual information according to the cognitive theory of multimedia teaching [4]. Thus, a minimalistic approach for the design is favorable, optimizing the students' ability to absorb the information presented. The design of the prototype has been centered around a minimalistic approach for these reasons.

Chapter 6

Future Work

There are several aspects to the developed prototype that needs to be addressed. These parameters are related to the teaching model derived from the framework proposed by Leftheriotis et al. [3]. Additionally, there exists alternatives for expansion in regard to social interactions, gamification and content creation. This chapter describes the necessary research needed for optimizing the efficiency of the application, and outlines possible expansions.

6.1 **Research**: Describes the necessary research needed for optimizing the teaching aspect of the application.

6.2 **Social Interactions**: Outlines possible expansions into integrated social interactions.

6.3 **Instructors**: Discusses functionality for instructors to create additional content and gain insights.

6.4 **Computer Graphics**: Adding more content specifically targeting students of computer graphics.

6.5 **Prototype**: A brief presentation of specific future improvements to the prototype.

6.1 Research

As discussed in Section 5.3, user testing should be postponed until a suitable context and time scope could be provided. This research should be specifically aimed at determining progression parameters. Specifically, this means obtaining data on how much time a student should spend in an activity before the system unlocks the next one, and if previous activities should be available once the next one has been unlocked. How many problems the student has to solve before unlocking the final stage should also be addressed.

The specialization project implied the use of click-streams for researching user patterns might be a viable approach [19]. While this approach might gain some viable insight at a later stage, it requires that functionality for generating click-streams are implemented. At this point, it is more useful to observe students using the prototype over time, and use more traditional collection methods such as questionnaires for collecting data. This approach targets the specific parameters better, and is more efficient. A proposed setup for the study is to divide students into groups, each group using different values for the parameters. To evaluate the parameters, the students' knowledge level should be tested before and after. This approach is similar to the approach of Leftheriotis et al. for testing the viability of the framework [3].

6.2 Social Interaction

Chapter 5 mentions the implementation differs from Leftheriotis et al. in that social interactions in the Application activity is absent. Expanding this activity to include features such as scores, specific problems (i.e. not randomly generated), and other forms of competition could prove advantageous in regard to learning outcomes. However, it is essential that such an expansion comes before the proposed study in the previous section. Altering the core function of one of the stages after such a study would possibly change the efficiency, and require additional research.

6.3 Instructors

Especially in regard to problem generation, allowing instructors to create templates for problems would increase the value of the experience for the students. More variety should decrease the possibility that students lose interest. Such an addition would not require significant efforts, especially compared to the potential gain.

As an addition to content creation, allowing instructors to follow the students' progression will potentially increase both the teaching and learning experience. This would create opportunities for instructors to catch struggling students, and possible take steps to resolve any problems they might have. However, other data-mining methods such as click-streams yield more powerful insights [20] Such methods would give instructors insights on advantageous use patterns [20].

6.4 Computer Graphics

Earlier attempts for teaching geometric transformations have incorporated more content specifically aimed at students of computer graphics. Specifically, I3T allows the user to investigate aspect such as view frustum, projection, view transformation, and more [1]. The AR tool had a function for automatically generating shader code [2]. Such elements were intentionally left out of the prototype, because a more minimalistic approach had been chosen.

Should adding such elements be considered in the future, adding them as separate sections is the best choice. Preserving the minimalistic approach also here, should help students understand the core concepts, without feeling overwhelmed by numerous configurable choices.

6.5 Prototype

There are numerous improvements that could be made to the prototype. Some of them more essential than others. Table 6.1 presents potential improvements and indicates their priority.

Table 6.1: Potential improvements to the prototype

Description	**Priority**
Function output as input to matrix positions with function input value manipulation	HIGH
Functionality for changing passwords, email, etc	MEDIUM
Two-factor authentication	MEDIUM
Require email confirmation on registration	MEDIUM
Functionality for instructors to design problem templates for the students	MEDIUM
Social interactions between students during application stage (e.g. high scores, competition)	MEDIUM
Functionality for instructors to see student progression	MEDIUM

Chapter 7

Conclusion

Teaching geometric transformations with the help of digital technology has been attempted several times. This thesis describes an alternative approach by using ideas from educational technologies, as well as elements from earlier work. The proposed teaching model is a derivation of a framework designed with educational theories in mind. A prototype has been developed based on this teaching model.

7.1 **Teaching Model**: Conclusions in regard to the teaching model.

7.2 **Prototype**: Reflections around the final prototype.

7.1 Teaching Model

The teaching model follows a minimalistic approach and is firmly rooted in educational technologies. Although the viability of the model has not been confirmed, the model it is derived from was based in epistemological traditions and preliminary results were positive. As outlined in 6, future research can confirm the efficiency of the teaching model.

7.2 Prototype

The resulting prototype is lean, and has the required functionality for further research. Development work has used industry standards to reach an MVP. While user testing has not been performed for general usability, the prototype should meet the minimal requirements as prominent frameworks and design methods were used. Future work on expanding the functionality should wait until the viability of the teaching model has been confirmed through research.

Bibliography

[1] I3t - an interactive tool for teaching transformations. `https://i3t-tool.org/`. (accessed: 04.06.2021).

[2] Thomas Suselo et al. Technologies and tools to support teaching and learning computer graphics. *ACE*, 2019.

[3] Letizia Jaccheri Ioannis Leftheriotis, Michail N. Giannakos. Gamifying information learning activities using interactive displays: an empirical investigation of students' learning and engagment. *Smart Learning Environments*, 4(1):2, 2017.

[4] R.E. Mayer. Cognitive theory of multimedia learning: The cambridge handbook of multimedia learning. *Why Learning Systems?*, 41:31–48, 2005.

[5] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: Defining gamification. volume 11, pages 9–15, 09 2011.

[6] Jihan Rabah, Robert Cassidy, and Robert Beauchemin. Gamification in education: Real benefits or edutainment? 05 2018.

[7] Theoharis T. et al. *Graphics & Visualization - Principles and Algorithms.* A K Peters Ltd, 2008.

[8] Petr Felkel et. al. I3t: Using interactive computer graphics to teach geometric transformations. *EUROGRAPHICS*, 2018.

[9] Introduction — material design. `https://material.io/design/introduction`. (accessed: 07.06.2021).

[10] Javascript. https://developer.mozilla.org/en-US/docs/Web/JavaScript. (accessed: 04.06.2021).

[11] Typed javascript at any scale. https://www.typescriptlang.org/. (accessed: 04.06.2021).

[12] Marble.js. https://docs.marblejs.com/. (accessed: 02.06.2021).

[13] Angular. https://angular.io/. (accessed: 02.06.2021).

[14] Opengl es for the web. https://www.khronos.org/webgl/. (accessed: 02.06.2021).

[15] The database for modern applications. https://www.mongodb.com/. (accessed: 10.06.2021).

[16] Introduction to json web tokens. https://jwt.io/introduction. (accessed: 09.06.2021).

[17] General data protection regulation. https://gdpr-info.eu/. (accessed: 10.06.2021).

[18] Webgl: 2d and 3d graphics for the web. https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API. (accessed: 05.06.2021).

[19] Daniel Lines. Teaching computer graphics transformations in webgl, 2020.

[20] Michail N. Giannakos et al. Making sense of video analytics: Lessons learned from clickstream interactions, attitudes, and learning outcome in a video-assisted course. *International Review of Research in Open and Distributed Learning*, 16:260–283, 2015.

Appendices

Appendix A

Acronyms

JWT — JSON Web Token

CTML — Cognitive Theory of Multimedia Learning

AR — Augmented Reality

TS — TypeScript

JS — JavaScript

UI — User Interface

API — Application Programming Interface

GPU — Graphics Processing Unit

GDPR — General Data Protection Regulation

Appendix B

Transformations

This section presents the matrices representing the Transformations in 3D. The material is gathered from [7].

$$\begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.1}$$

Matrix (B.1) is a translation transformation where (a, b, c) represents how much the vector $v \in \mathbb{R}^3$ will be translated in respective directions x, y, z.

$$\begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.2}$$

Matrix (B.3) is a scaling transformation where (a, b, c) represents how much the vector $v \in \mathbb{R}^3$ will be scaled in respective directions x, y, z.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.3}$$

Matrix (B.3) is a rotation transformation where θ represents how much the vector $v \in \mathbb{R}^3$ will be rotated around the x-axis.

$$\begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & \cos\theta & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.4}$$

Matrix (B.4) is a rotation transformation where θ represents how much the vector $v \in \mathbb{R}^3$ will be rotated around the y-axis.

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.5}$$

Matrix (B.5) is a rotation transformation where θ represents how much the vector $v \in \mathbb{R}^3$ will be rotated around the z-axis.

$$\begin{bmatrix} 1 & c & e & 0 \\ a & 1 & f & 0 \\ b & d & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.6}$$

Matrix (B.6) is a shearing transformation where (a, b, c, d, e, f) represents how much the vector $vin\mathbb{R}^3$ will be sheared in directions (x, yz). $(a, b), (c, d), (e, f)$ corresponds to shearing in respective directions x, y, z.

Appendix C

Prototype Sketches

This appendix presents sketches made early in the process during the specialization project.

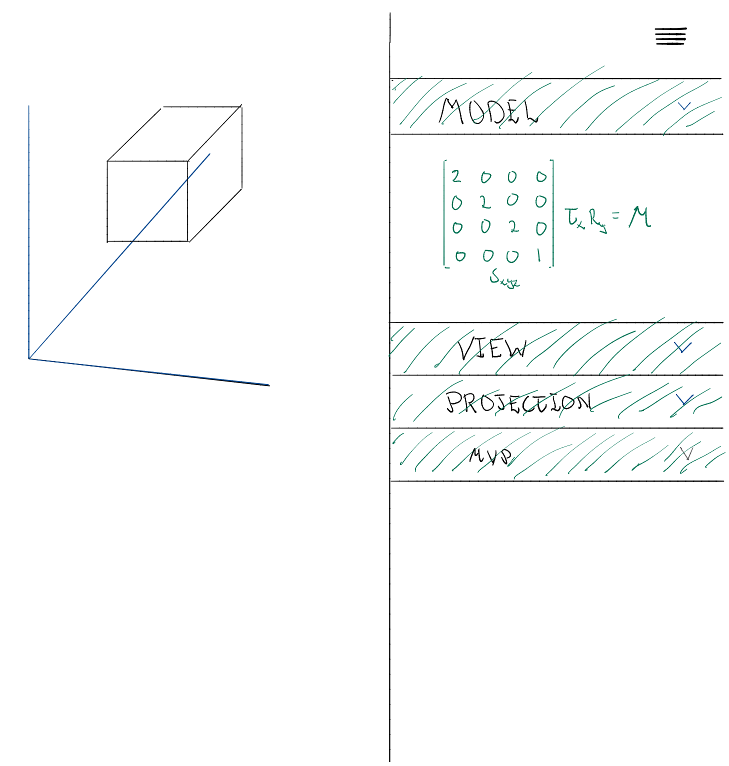

Figure C.1: Sketch of UI manipulating matrix for transformation in model space

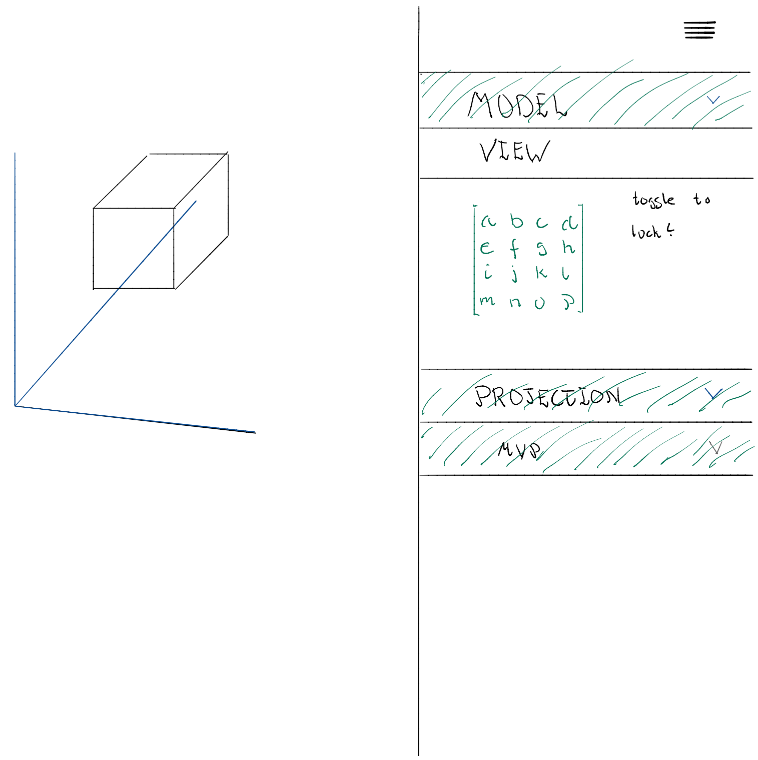

Figure C.2: Sketch of UI manipulating matrix for transforming to view space

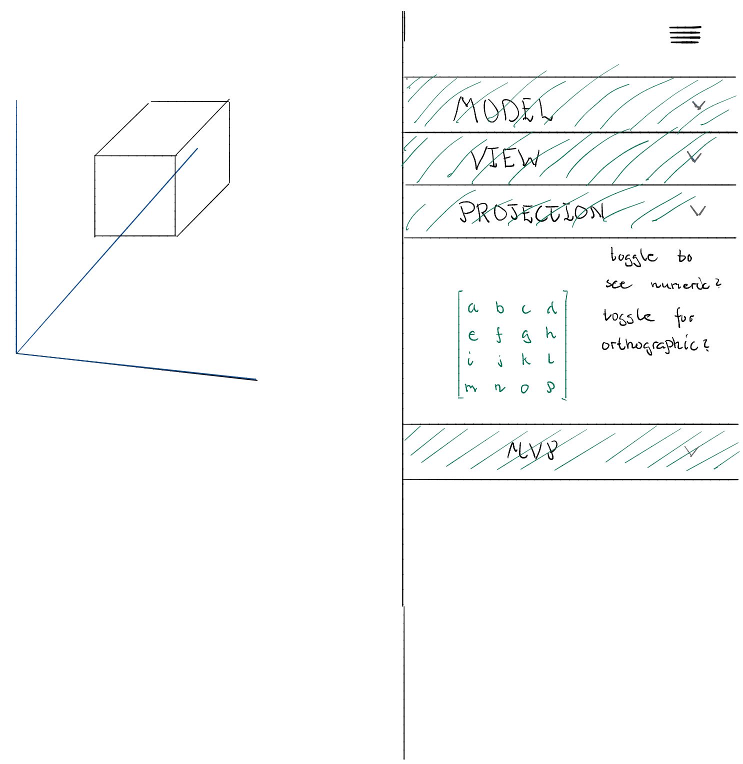

Figure C.3: Sketch of UI manipulating perspective projection matrix

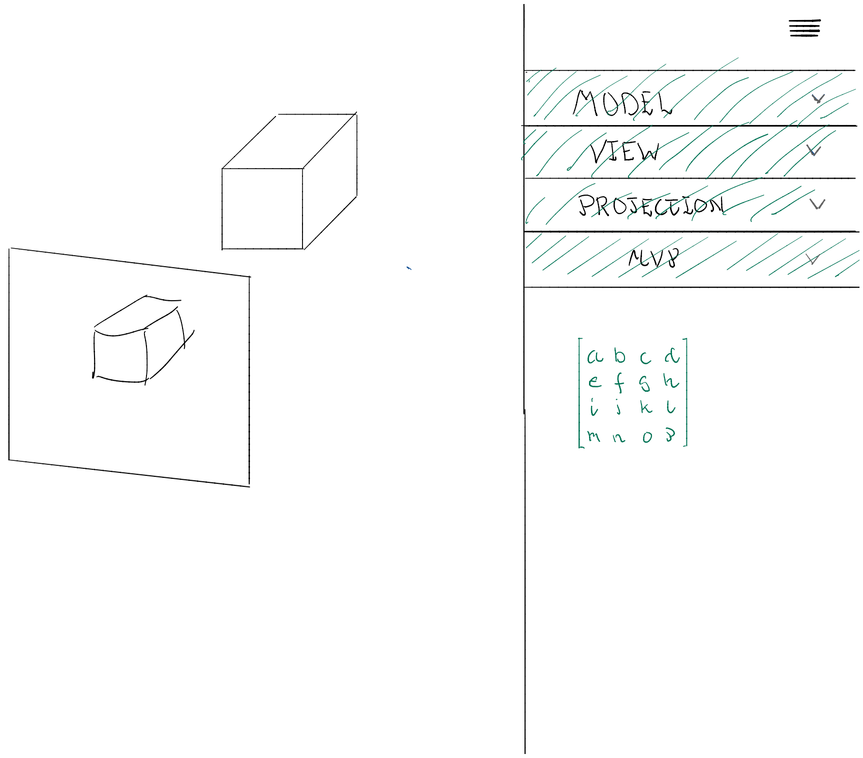

Figure C.4: Sketch of UI showing the plane with the projection and the calculated MVC matrix

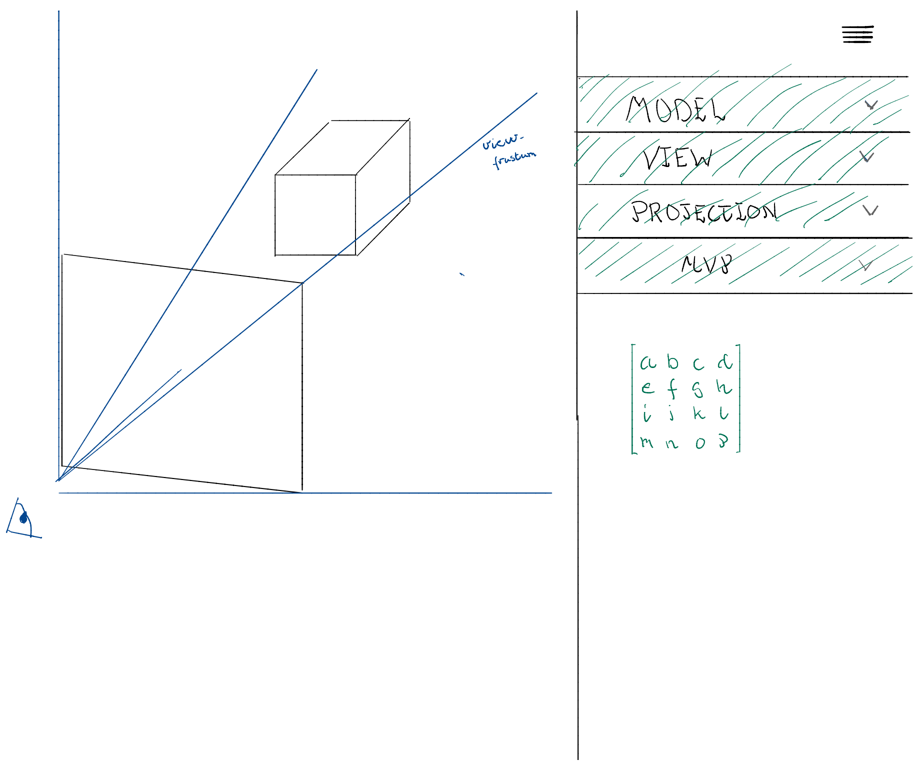

Figure C.5: Sketch of ideas for UI with orientation information of the view frustum and projection plane

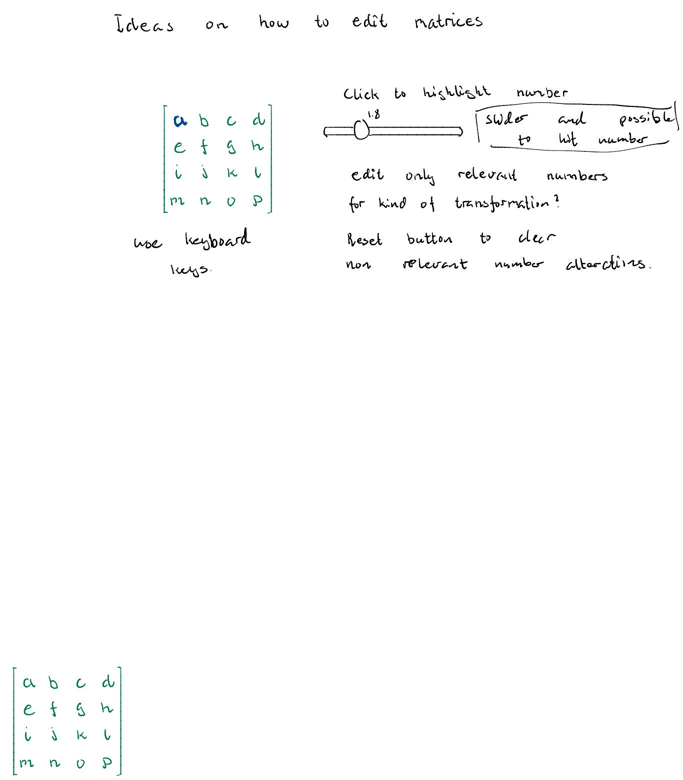

Figure C.6: Sketch with ideas for interaction with matrices

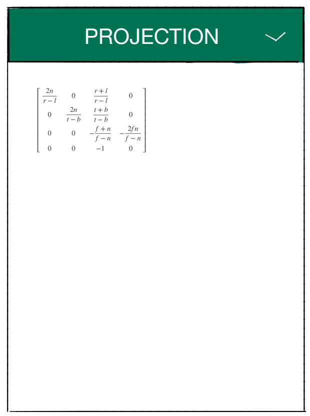

Figure C.7: Sketch of ideas for matrix representation in the UI

Appendix D

Install Instructions

The following is a guide for installing the necessary dependencies and running the prototype.

uTransform

Teaching geometric transformations using modern web technologies. The prototype consists of a back-end API and a front end application. Both have to be running for the prototype to work correctly.

Dependencies

The dependencies for each of the two components are listed in their respective README.md.

Running the prototype

To run the prototype, follow the instructions in the two README.md files.

Constraints

The requirements for accessing the different activities are defined in the front-end code. This definition is located in `uTransformClient/src/constants.ts`. For demonstration purposes, they have been set to zero at this time. This means that a user will have access to all of the activities once registered.

To reactivate this feature, change the value of the thresholds in `constants.ts` named `timeThreshold` and `problemThreshold`.

uTransformApi

Back-end API for the uTransform prototype.

Dependencies

The following dependencies are necessary for running the API,

1. Node.
2. NPM.
3. MongoDB.

Installing packages

After installing the dependencies listed above, run `npm install` in the uTransformApi directory. This will installed the necessary JavaScript packages to run the application.

Run

Run `npm run start` to start the API.

uTransformClient

Front end for uTransform - a prototype for teaching geometric transformations.

Dependencies

Node and npm needs to be installed on the computer attempting to run.

Install packages

After installing node and npm run `npm install`. This will install the necessary packages

Run

Run `ng serve`. The front end application is now available on localhost:4200.

NTNU
Norwegian University of
Science and Technology