Multidiszciplináris tudományok, 13. kötet. 3. sz. (2023), pp. 241-256 **https://doi.org/10.35925/j.multi.2023.3.24**

AN EVOLUTIONARY APPROACH TO DEVELOPING SUPPORTING SOFTWARE FOR THE DESIGN OF CARD DECK-BASED MATHEMATICAL DIDACTIC GAMES

Enikő Palencsár
BSc student, University of Miskolc, Faculty of Mechanical Engineering and Informatics
3515 Miskolc-Egyetemváros, e-mail: palencsar.eniko@student.uni-miskolc.hu

Szilvia Szilágyi
associate professor, University of Miskolc, Institute of Mathematics, Department of Analysis
3515 Miskolc-Egyetemváros, e-mail: szilvia.szilagyi@uni-miskolc.hu

Abstract
Didactic card games have demonstrated their efficacy in engaging Generation Z students in diverse course materials in the field of natural sciences. However, the absence of suitable tools poses a significant obstacle to the creation of card decks which include complex formulas. In this paper, we introduce a JavaScript-based web application that simplifies and accelerates the design process by providing support for LaTeX that enables the integration of scientific equations onto the cards. The proposed software aims to encourage educators to participate in card deck design, with the ultimate goal of contributing to the broader adoption of scientific card games.

Keywords: *game-based learning, card games, didactic games, educational technology, game design, application development*

1. Introduction

Game-Based Learning – or GBL in short – gains more and more traction nowadays, given that Generation Z students, born in the mid to late 1990s, generally prefer an interactive, experiential, varied learning experience as opposed to traditional classroom-environment courses (Hernández de Menéndez et al., 2020). Game-Based Learning refers to an environment where games enhance the acquisition of knowledge and skills, while providing players/learners with problem-solving opportunities and a sense of achievement (Qian & Clark, 2016). In today's education, GBL takes shape through two primary formats: digital, involving computer games played on IT devices, and non-digital, represented by traditional board games (Santos, 2017). Both have their unique advantages, the choice between them depends on the specific learning goals, the learning environment, and the preferences of the players.

Computer games offer rich and immersive multimedia experiences, incorporating visuals, animations, and sound effects, creating a captivating learning environment that resonates with Gen Z students, often referred to as "Digital Natives" due to their exposure to rapid technological advancements from an early age (Moore et al., 2017). Computer games are excellent at adapting to individual players' abilities, providing personalised learning experiences and a possibility for skill-based differentiation, as well as tailoring the gameplay to the needs of unique learners. Another strength of computer games is their ability to provide immediate feedback, a characteristic highly valued by Gen Z students (Isaacs et al., 2020). Digital GBL offers timely insights on performance, highlighting areas of improvement and rewarding achievements. This prompt feedback loop motivates players and helps them learn from their mistakes more efficiently.

On the other hand, physical board games offer a tangible learning experience, allowing students to engage in hands-on, tactile interactions with the game components. They are generally considered cost-effective instructional materials (Santos, 2017). In addition, they encourage face-to-face interaction with other players, promoting communication, collaboration, and teamwork. Unlike computer games, physical board games can be utilised in environments like traditional classrooms where electronic devices are not readily available, eliminating the need for Bring Your Own Device (BYOD) policies. In many Hungarian educational institutions, especially in primary and secondary schools, the availability of IT devices is often limited. As a result, traditional board games have become the central focus of our didactic game development research, offering an accessible and effective means of incorporating interactive learning into these educational settings.

Through our extensive research conducted over the past few years (Árvai-Homolya et al., 2019; Lengyelné Szilágyi et al., 2019; Szilágyi & Körei, 2022; Dudás et al., 2019), we have discovered that card deck-based games offer a suitable framework for designing diverse educational games with opportunities for differentiation. The versatility of card decks lies in their simplicity and affordability, as, contrary to metal figurines or plastic game components, they can be easily printed onto sheets of paper, even at home. Manipulating and redesigning cards is also straightforward, allowing for flexible game design iterations. Moreover, card decks provide the flexibility to create multiple decks for the same didactic game, gradually increasing in difficulty to better adapt to individual student needs and skill levels. A card deck introduces an element of randomness and variety into the gameplay, improving the replay value of the game. Furthermore, card decks are typically compact and portable, facilitating easy transportation to different locations such as classrooms, homes, or outdoor environments, thus allowing for flexible learning opportunities.

At the University of Miskolc, various didactic games were developed in the past, particularly as a supplement to the course materials of Mathematical Analysis I and II. For instance, the card game LimStorm, strongly based on the popular game SOLO was constructed to help students understand the concept of limits (Szilágyi & Körei, 2022), Blue Yeti serves as practice for the comparison test of improper integrals, and Gem Hunters takes players on a tropical adventure with a special focus on trigonometric functions (Dudás et al., 2019). All these games employ a card deck specifically designed for educational purposes, incorporating various different mathematical formulas.

2. State of the Art

The need for a flexible, reusable card deck designer software became evident following the development of two desktop applications supporting the design of cards specific to the games Gem Hunters (Dudás et al., 2019) and LimStorm (Szilágyi et al., 2023). By that point, it was apparent that more didactic card games would be created in the future. However, the development of new, separate applications for each game would be a repetitive, inefficient, and costly endeavour. Instead, our focus shifted towards the creation of a universal and multipurpose card deck editor application that can be used in the context of various scientific didactic games even beyond the field of mathematics. The primary requirement for this software was to provide LaTeX support, enabling the efficient and convenient placement of equations onto the cards.

2.1. The significance of LaTeX

LaTeX is a typesetting system and markup language used for the creation of high-quality documents, particularly those involving complex mathematical equations (Lamport, 1994). It was developed in the 1980s by Leslie Lamport and is based on the TeX typesetting system created by Donald Knuth (Merriam, 2023). Unlike conventional word processors, LaTeX focuses on the structure and formatting of documents rather than their visual appearance. Users write LaTeX code in plain text, which is then processed by a LaTeX compiler to produce a well-formatted document. This separation of content and typesetting allows writers to focus on the content while leaving the document layout and design to LaTeX.

LaTeX provides a wide range of features for producing professional-looking documents. It offers extensive support for mathematical equations and symbols, automatic numbering of sections and figures, bibliography management, cross-referencing and more (LaTeX Documentation). It is widely used by academic researchers, especially in fields such as mathematics, computer science, engineering, and other technical disciplines. It is known for its ability to handle complex mathematical notation and its precise control over document layout.

$$\int\limits_{1}^{\infty} \frac{\pi}{2x^3}\, dx \tag{1}$$

Let's consider for instance the improper Riemann integral presented in *Equation (1)* In LaTeX notation, this integral can be expressed as *\int\limits_{1}^{\infty} \frac{\pi}{2x^3} \mathrm{d} x*, using plain text with characters that can be found on most standard keyboards. Therefore, the support of LaTeX notations in the editing process of didactic card decks provides a simple, standardised way of adding scientific formulas to the cards. This highlights the pivotal role of LaTeX in making a card editor for didactic games truly universal.

2.2. Alternative approaches to card editing

Two major alternatives arise to developing our own software tools when it comes to the design of card decks for game-based learning purposes. Traditional image editors, such as Adobe Photoshop offer a wide array of services, ranging from a simple crop or resize to merging pictures together. However, these editors, commonly used for general graphic design purposes, may not be well-suited for card deck editing due to their inherent limitations, lacking features tailored specifically to the design of card decks. As an example, traditional image editors do not provide a comprehensive view of the entire card deck, making it challenging for users to assess the overall design coherence and consistency. Unlike dedicated card deck designers, traditional image editors may not offer an option to export the card deck in a format optimised for printing. Users might encounter difficulties in preserving the correct dimensions, resolution, bleed lines, and other specifications essential for professional printing. Consequently, although traditional image editors are versatile tools for general graphic design purposes, their limitations make them less suitable for efficient and streamlined card deck creation.

On the other hand, online card editor tools are specialised in the creation and customisation of card decks with convenient and user-friendly interfaces and a range of design options. Their most important drawback – that applies to image editors as well – is that online card deck designers generally do not provide support for incorporating scientific formulas onto the cards. While they offer extensive customisation options for graphics, text, and layout, the inclusion of complex mathematical expressions may not be accommodated within their design interfaces. Moreover, these applications are often associated with specific service providers, to the facilities of which the printing of the designed card decks is typically limited (e.g.: makeplayingcards.com). This lack of interchangeability means that users are bound to the specific provider whose designer they used, the designed decks cannot be exported or saved for independent printing.

Table 1. *The advantages and drawbacks of the two main alternative solutions*

<table>
<tr><th></th><th>Image editors</th><th>Online card editors</th></tr>
<tr><td>Advantages</td><td>diverse features</td><td>specialised features</td></tr>
<tr><td rowspan="3">Drawbacks</td><td>absence of an overall view of the card decks</td><td rowspan="3">often connected to a service provider (e.g.: printer studio, online gift shop) or a specific game with given design elements</td></tr>
<tr><td>no built-in formatting for printing</td></tr>
<tr><td>often complicated to use</td></tr>
<tr><td>LaTeX support</td><td>none</td><td>none</td></tr>
</table>

To overcome the limitations of traditional image editors and online card deck editors, presented in *Table 1*, it becomes necessary to develop an application that integrates the advantages offered by both and provides support for adding LaTeX equations onto the cards. By merging the strengths of the aforementioned tools, the resulting application can provide a comprehensive and user-friendly solution for efficient card deck creation in educational contexts. Such an application offers heightened flexibility, a comprehensive view of the deck, and export options with optimised print-ready outputs.

3. Methodology

In the field of software engineering, multiple models of development have been devised to guide the creation and delivery of software systems. These models offer different approaches and methodologies, each with its own set of principles and advantages, allowing developers to choose the most suitable model based on project requirements and constraints.

The incremental model of software engineering is an iterative approach that builds upon the foundation of the waterfall model. The waterfall model, a linear sequential approach, follows a strict sequence of phases such as requirements specification, design, implementation, validation, and maintenance (Adenowo & Adenowo, 2013). However, the use of a top-down approach in the waterfall model limits customer involvement in the software development process. As a result, evolutionary process models, such as incremental implementation and prototyping, have emerged to address this issue (Denning et al., 2008).

The incremental model divides the software development process into smaller increments, also referred to as iterations. Each iteration involves the execution of all phases of the waterfall model but applied incrementally to a subset of the requirements (Ruparelia, 2010). In the incremental model, the development process begins with an initial subset of requirements, which undergoes the complete waterfall lifecycle from requirement analysis to maintenance. Once this subset is developed and delivered, subsequent increments are added, building upon the previous functionality. Each iteration refines and expands the software system until it reaches the desired level of completeness *(Figure 1)*.

The incremental model offers several advantages. Firstly, it allows for early and continuous delivery of working software, enabling stakeholders to give feedback and make adjustments throughout the development process. This iterative nature provides flexibility, adaptability, and customer satisfaction. Secondly, the incremental model reduces the risk associated with large-scale development projects, as issues can be identified and addressed early on in smaller increments. Additionally, it enables concurrent development of different system modules or components, fostering collaboration among development teams (Kneuper, 2018).

On the other hand, the incremental model also has some considerable drawbacks. The need for multiple iterations and ongoing integration can increase the complexity of project management, given that the system architecture must be carefully designed to accommodate future increments without sacrificing scalability and maintainability. Moreover, the incremental model requires effective communication and coordination between stakeholders and development teams to ensure alignment on priorities and requirements (Petersen & Wohlin, 2009).

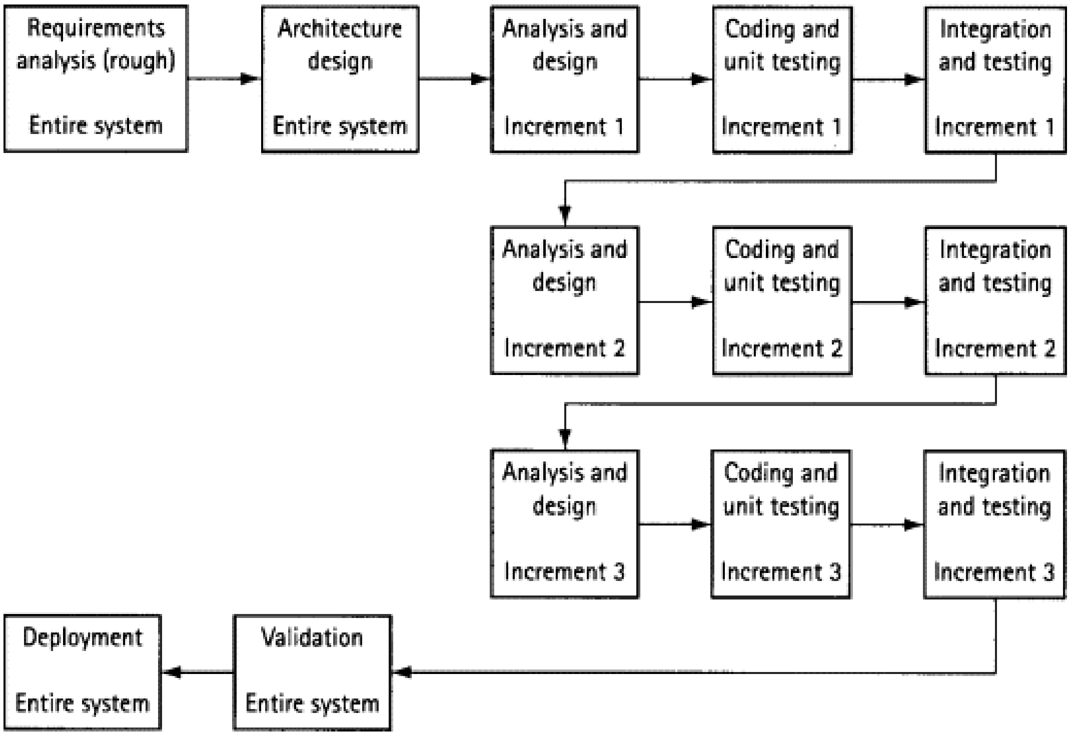

Figure 1. *An incremental process*
Source: https://www.sciencedirect.com/topics/computer-science/incremental-development

In the particular case of the Card Deck Designer, given the limited initial information available regarding the desired outcomes, a deliberate choice was made to employ the incremental model of software development. This methodology entailed the development of a functional evolutionary prototype as the foundation, followed by iterative adjustments and additions to the web application based on the emerging requirements in subsequent development stages.

4. Requirements analysis

In the requirement analysis phase of the software development, the basic functions and restraints of the application were established. Our primary aim was to develop a universal, user-friendly, and fully functional web application for the design and editing of card decks used in mathematical didactic games. To ensure the success of this endeavour, we carefully examined existing tools for card deck design and engaged in consultations with educators who have experience using or expressed interest in incorporating didactic card games into their teaching practices. The criteria established in the earlier stages of development, to serve as directives for the first steps of the software process, were as follows:

1. The application should be readily available and usable on most devices, preferably without need for downloading and installing any additional files. Hence, the idea to build a web application was born.
2. Support for compiling LaTeX formulas and seamlessly integrating the resulting equations onto the cards should be provided.
3. Users should be provided with the capability to view the entire card deck on a single page, allowing them to observe and assess the design coherence and consistency of the deck as a whole.
4. The application should also provide flexible options to export card decks in a format optimised for printing.

The identified criteria formed a strong foundation for the initial version of the Card Deck Designer, driving the first iterations of the development process. As the implementation and validation of the application progressed, further important constraints and services were identified, leading to subsequent increments in the iterative process.

4.1. Stakeholders

In the software development process, a stakeholder is a party affected by the system being developed, who has the ability to influence the requirements. By actively involving the stakeholders throughout the development process, the software can be tailored to address their unique needs and deliver an improved user experience. In the case of the Card Deck Designer, which is considered an off-the-shelf software, designed to meet the requirements of a diverse user base, the key stakeholders are the developers and the potential end users, specifically teachers and lecturers who are interested in incorporating thematic, uniquely designed card games into their lessons.

4.2. Constraints

During the development of the Card Deck Designer, various constraints were encountered. One of the primary constraints was the limited time frame of a few months within which the application needed to be completed. This time constraint was imposed by the need to create and design card decks for our own mathematical didactic games. Another constraint that had to be addressed was the need for wide availability and platform independence, the web application had to be accessible across different devices and operating systems. This constraint required careful consideration of cross-platform compatibility, responsive design principles, and thorough testing to ensure compatibility across various devices and browsers.

Additionally, scalability and performance were critical factors in the development process. It was important to ensure that the application could handle larger card decks without compromising on response times. As highlighted by Jakob Nielsen (Nielsen, 2010), there exist 3 major response time limits that developers need to be aware of: 0.1 seconds creates the illusion of an immediate response, giving users the perception that their actions directly influence the outcome. Within 1 second, users maintain a seamless flow of thought, recognising a slight delay but still feeling in control of the overall experience. Within 10 seconds, users remain engaged, although they begin to feel dependent on the computer and desire faster responses. However, a 10-second delay often leads users to leave a site immediately.

4.3. Functions and services

We note that all the card deck elements showcased in the figures of this paper are components of the YETI educational framework, the graphics of which were developed in parallel with the Card Deck Designer application. These elements include three distinct cardback images featuring the Koch snowflake *(Figure 2)*, as well as multiple different yeti and avalanche cards. Additionally, an edge pattern, also inspired by the Koch curve, was designed for the cards, presented in two different colours to symbolise convergence and divergence *(Figure 3)*.

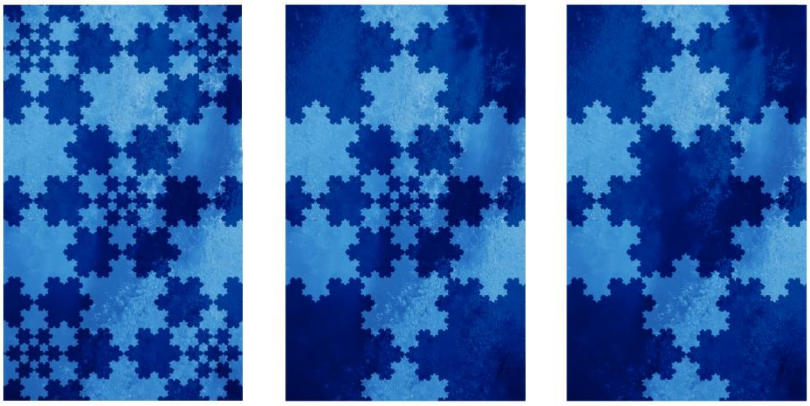

Figure 2. *YETI cardback images for three levels of difficulty: expert, intermediate and beginner (left to right)*

In the subsequent sections, the primary features and functionalities offered by the Card Deck Designer are presented.

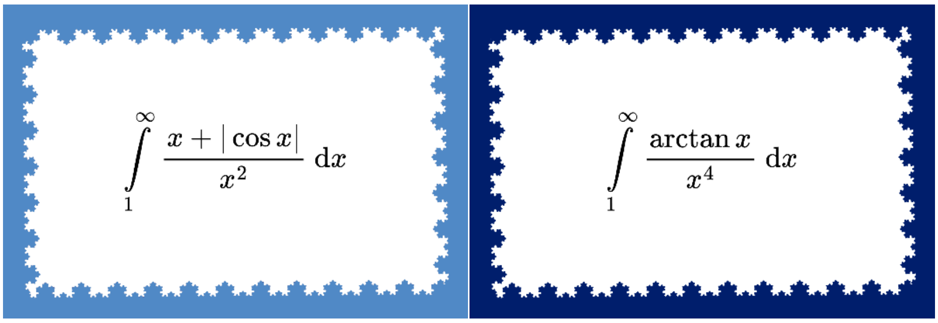

Figure 3. *Cards from the game Blue Yeti (part of the YETI framework) featuring edge decorations in different colours*

4.3.1. Configuration Options

The Card Deck Designer encompasses several important features to provide personalised editing options *(Figure 4)*. This flexibility ensures that the software can be adapted for various purposes, accommodating a wide range of applications. In terms of card size, the designer allows users to apply custom dimensions to all cards in a deck. Although the ratio of the two sides is fixed at 0.65 to prevent cropping issues, the cards' width can be customised between 50 and 100 millimetres.

Furthermore, users can select and upload an image from their device, which will serve as the cardback design for all cards within the same deck. The designer includes options to resize and crop the uploaded image, ensuring it is properly fitted onto the cards. This functionality enables users to create a cohesive visual identity for the entire deck.

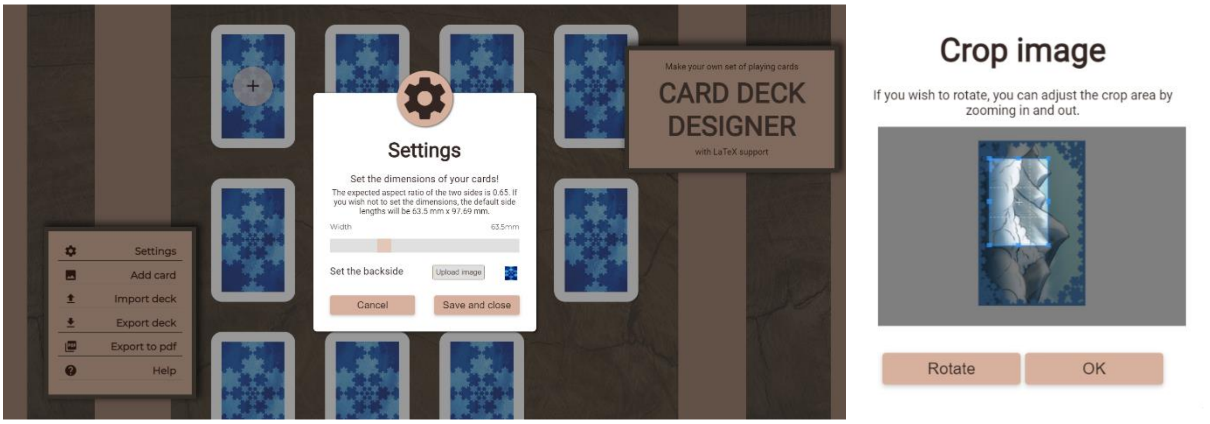

Figure 4. *The settings window and the cropping window of the application, the latter featuring an avalanche card from the YETI deck*

4.3.2. LaTeX formula handling

Users of the Card Deck Designer have the ability to create cards with unique background images, onto which they can later place scientific formulas. Within the Add card menu, users can select a background image of their choice and specify the number of cards to which they want to apply that specific background. Initially, 10 card slots are available on the screen, but this capacity can be expanded to a maximum of 150 cards by uploading additional designs. However, to prevent potential performance issues, it is advisable for users desiring to edit decks with a larger number of cards to partition their development into smaller sections.

In addition, the Card Deck Designer offers the vital functionality of incorporating mathematical formulas onto the cards. Users can input LaTeX expressions into a designated text field, which are then compiled and precisely positioned onto the card *(Figure 5)*. The application provides the flexibility to adjust the vertical position of the equation, while maintaining a default centred alignment horizontally. The equation can also be resized to achieve the desired visual effect.

Furthermore, the application allows users to manage modifications effectively. They have the option to reset all changes made, reverting the card to its original state. Alternatively, they can save their progress, capturing all adjustments and additions made to the card. It is important to note that saved changes are definitive, and once a card is saved with a formula on it, subsequent resets will not be possible.

Figure 5. *The card editing window*

4.3.3. Export Options

The Card Deck Designer offers various options for saving and exporting a deck of cards. Users have the ability to save individual cards as PNG files, allowing for the sharing of each card as a separate image. Additionally, the application provides an export-to-PDF feature that allows users to compile their entire card deck into a PDF document. The popup menu for export offers a brief recapitulation of the current settings in place and two distinct export options:

1. Exporting the deck with one card per page in PDF format, suitable for professional printing. This option ensures that each card is presented on a separate page. Notably, the cardback design is also presented on a separate page, positioned at the end of the deck. This approach facilitates high-quality printing.
2. Printing on an A4 sheet of paper, with the ability to set margins between and around the cards. Users can customise the margins within the range of 0 to 30 millimetres, providing flexibility for printing at home. This option caters to those who prefer to print the cards on their own, enabling precise control over the spacing and alignment of the card designs on the printed sheet. In this case, the pages of the document are alternating between card fronts and card backs with a precise alignment, suitable for double-sided printing. As many cards as possible are placed on each A4 page, ensuring optimal use of space.

4.3.4. User Manual

The Card Deck Designer incorporates an informative Help menu, providing users with access to a comprehensive user manual that assists them in navigating the website and utilising its features effectively. The Help menu is designed to address any questions or uncertainties users may encounter during their interaction with the website, divided into seven functionality-based sections. Important notes and warnings are highlighted by a different coloured background at the end of each section.

5. Implementation

For the development of the web application, a strategic decision was made to utilise HTML, CSS and JavaScript as the core technologies. HTML (Hypertext Markup Language) forms the foundation of web development, providing the structure and semantic markup for web pages. It offers a robust and widely supported platform for creating the user interface and organising the content of websites. With its simple syntax and extensive browser compatibility, HTML was a natural choice to ensure broad accessibility and consistent rendering across different devices and browsers.

CSS (Cascading Style Sheets) was chosen for its powerful styling capabilities. It enables the precise control and customisation of the visual appearance of a web application. The design elements of the Card Deck Designer software such as colours, fonts, and layout were implemented using CSS, ensuring a visually appealing and user-friendly interface. In the development process, a decision was made to use native CSS rather than relying on popular frameworks (Bootstrap, Foundation…etc.), as native CSS generally allows for greater control and customisation over the styling of websites.

JavaScript, as a versatile programming language, was selected for its ability to add interactivity and dynamic functionality to the web application. With JavaScript, responsive behaviour could be implemented smoothly, and it also allowed for the integration of several useful external libraries.

5.1. External libraries

In the development process of the web application, three open-source external JavaScript libraries were used for the efficient implementation of specific key features.

1. MathJax (version 3.2.2): This library facilitates the creation of mathematical formulas within the application by transforming LaTeX expressions into SVG images.
2. Cropper.js (version 1.5.12): Used for image cropping, Cropper.js proves essential in ensuring that images are cropped to the correct aspect ratio.
3. jsPDF (version 2.5.1): Another significant library employed in the project is jsPDF, which facilitates the rendering of simple PDF documents containing card decks.

5.2. Design considerations

The design and visual aesthetics of the application were inspired by the concept of old wooden card-tables, aiming to evoke a sense of nostalgia. The main page, featuring a cracked wooden background, serves as a visual metaphor reminiscent of a worn-out table where the card deck being edited is laid out. To maintain focus on the cards themselves, a predominantly neutral colour palette was chosen for the website, consisting of browns and beiges. The current cardback image serves as a placeholder for cards that do not yet have uploaded front sides. The first card on the page acts as a shortcut for the Add cards menu and provides a visual representation of the currently set backside, even if all available card slots are already filled. The development of the popup windows for different menus prioritised simplicity and functionality, incorporating input fields for text and images, buttons, and sliders for convenient user interactions.

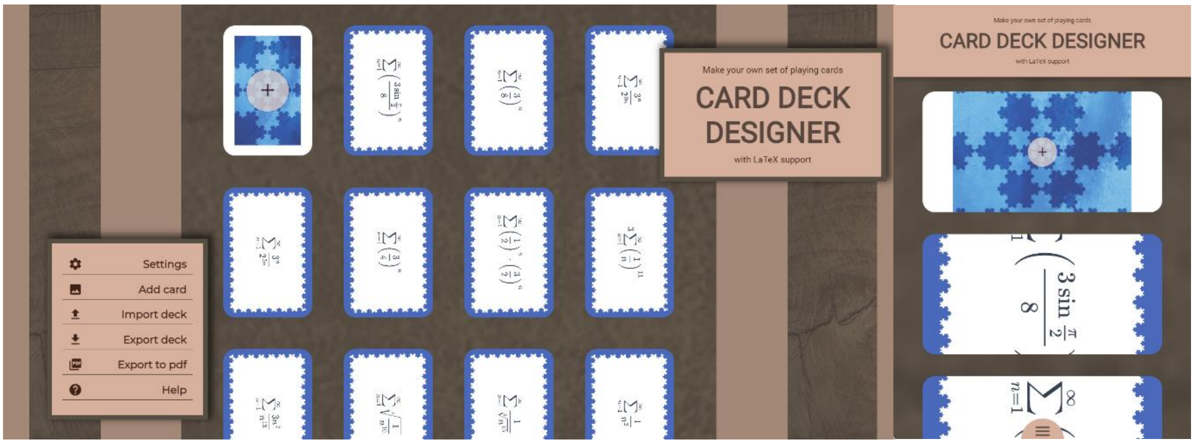

Figure 6. *The main page of the application on different devices*

The layout of the page is fully responsive, adapting to the dimensions of the viewer's screen. The number of cards displayed in each line of the main grid, as well as the positioning of the main title and the menu, adjust dynamically *(Figure 6)*. This responsive design approach ensures that the application is equally suitable for both desktop and mobile usage. On mobile devices with a narrow screen, the main menu transforms into a separate popup accessible by tapping the hamburger icon located at the bottom of the website.

6. Deployment and validation

In accordance with the incremental development approach, each implemented increment undergoes a comprehensive validation process and integration with the existing components of the application. The validation comprises both dynamic elements, such as testing procedures to assess the functionality in action, and static elements, including an overview of the requirements and the source code.

The initial working draft of the Card Deck Designer, encompassing all the essential and high-priority functionality, was completed by September 2022. Subsequently, the development process continued in smaller increments, with each iteration focusing on the realisation of specific additional services or addressing any bugs discovered during testing.

The validation process of the Card Deck Designer involved thorough functional testing of the different menus within the application. Additionally, the system underwent a series of acceptance tests that encompassed the editing of card decks specifically designed for the card games YETI and Blue Yeti. The main features and the responsiveness of the web application were continuously tested across multiple browsers, including Microsoft Edge and Google Chrome, as well as on Android devices, ensuring compatibility and optimal performance across different platforms and devices.

6.1. Acceptance tests

YETI is an educational board game designed to familiarise university students with the concept of infinite series. The game's conception was rooted in the outcomes of a comprehensive three-pillar survey conducted among bachelor students in the field of IT (Palencsár & Szilágyi, 2023). The game features a deck of 100 cards, 80 of which incorporate unique infinite sums and 20 additional cards with original designs, consisting of 16 yeti cards and 4 avalanche cards. The development of the game began in the autumn of 2022 and is currently ongoing. To cater to different levels of expertise, three distinct card decks—beginner, intermediate, and advanced—are being produced, accommodating students with varying skills and understanding of the given topic.

During the autumn and winter of 2022, beta tests of the intermediate YETI card deck were conducted with the assistance of the Card Deck Designer, ensuring continuous utilisation and in-practise testing of the application ever since its first iteration was completed. In the winter of 2023, the structure of the YETI game and the intermediate deck were finalised, paving the way for the first actual acceptance tests. The Card Deck Designer proved valuable in editing the thematic deck of 100 cards, the contents of which had been prepared over several months, within a span of approximately 90 minutes, without need for any additional tools or supplies. The efficiency of the export process remains a critical performance factor. As seen in *Table 2*, in the case of the YETI deck, both PDF export options took around 90 seconds to complete.

Table 2. *The export times of the two card decks, using different export options* (Google Chrome)

		Export time (seconds)	
Game	Number of cards to export	One/page	Multiple/page
YETI	100	93	93
Blue Yeti	29	39	37

In April 2023, our focus briefly shifted to the development of another game named Blue Yeti, inspired by the traditional card game Old Maid but with a unique twist to incorporate improper integrals into the gameplay. The Blue Yeti deck consists of 28 cards featuring integrals and one additional card depicting the Blue Yeti. Once again, the Card Deck Designer was employed for assembling and exporting the specialised card deck *(Figure 7)*, the export taking a little less than 40 seconds in this case.

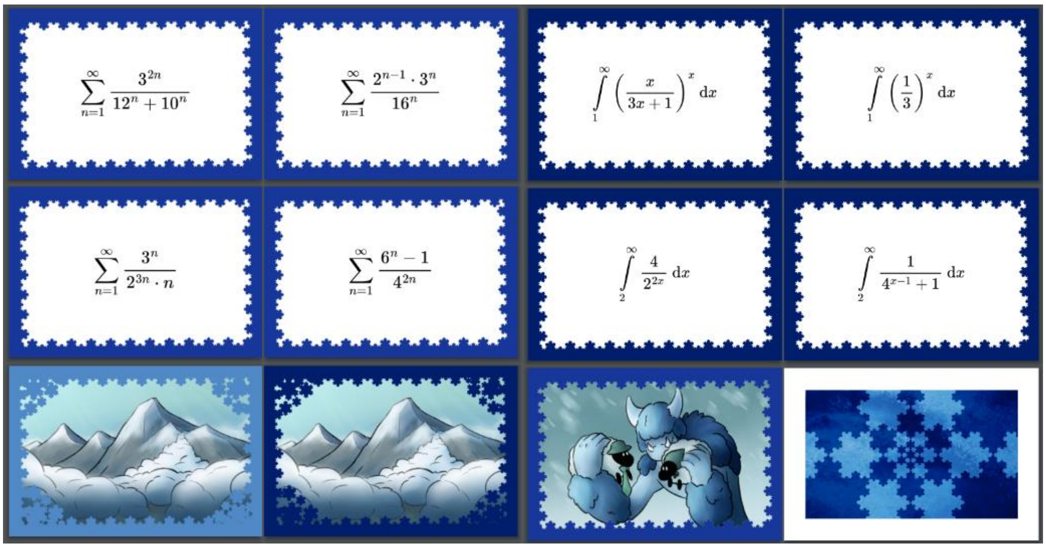

Figure 7. *Cards from the YETI and Blue Yeti decks, created using the Card Deck Designer*

The continuous testing of the web application unveiled numerous additional requirements that served as guidelines for subsequent increments in the development process.

6.1.1. Card deletion

In the initial iteration of the application, the option to delete specific cards from a deck was absent. As a result, users faced challenges when attempting to convert only a portion of their designed card deck to a printable format. They had to resort to either restarting the editing process for a smaller deck or individually saving the chosen subset of cards in PNG format and subsequently re-uploading them into a new deck. Recognising the significance of this missing feature, it was imperative to incorporate it, making it possible to delete a card by the press of a button in the editing window.

6.1.2. LaTeX help

To further support the universal usability of the Card Deck Designer, it was important to consider the diverse range of potential users, including teachers and lecturers across various educational levels, from primary school to higher education. It is important to note that not all educators are familiar with LaTeX notations, which could discourage them from using the application. Consequently, in the second iteration of the application, a LaTeX help window was introduced, presenting the most important LaTeX commands and notations that can assist in the design of mathematical card decks.

6.1.3. Counter for PDF export

The addition of a counter to the PDF export window is particularly beneficial for card decks consisting of 30 or more cards – such as the deck for the game YETI –, for which the export delay can exceed 30 seconds. The counter provides real-time feedback to users, thus shortening the response time and indicating the progress of the export process. In the case of the one-card-per-page rendering, it displays the number of cards exported, while for the A4-format export, it reflects the number of completed pages *(Figure 8)*.

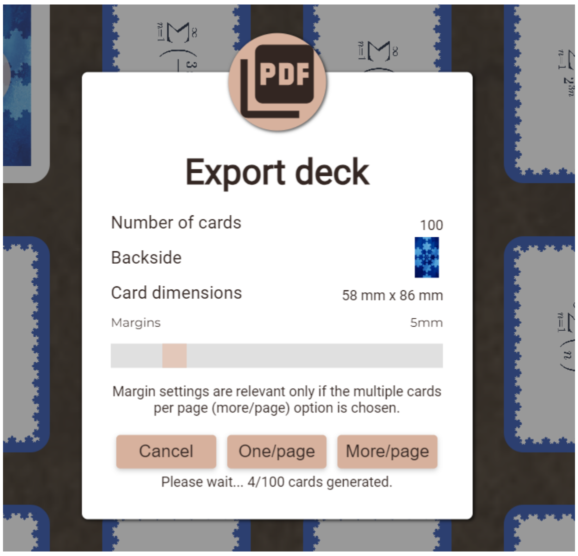

Figure 8. *The export menu of the application. The counter can be seen at the bottom of the window*

6.1.4. Flipping cards

During the editing of the card decks for the aforementioned games, it became evident that for certain didactic games, involving longer and more complex formulas, a horizontal orientation of cards would be more suitable. This preference came up in connection with both the game YETI and Blue Yeti.

Consequently, an orientation menu was added to the LaTeX editing window, enabling users to select whether they wish to place equations horizontally or vertically on the cards.

6.1.5. Free Movement and Background Customisation for Formulas

As an additional customisation feature, users now have the option to freely position formulas on the cards by dragging and placing them into their desired location. This enables greater flexibility in designing the card layout. To further extend the customisation options available to users, another new feature was introduced, allowing formulas to appear either on a transparent or a white background.

6.1.6. Text Files for Import and Export

To make the editing process more flexible, a new service was implemented that saves the image data associated with all cards, along with the deck's configuration, into a text file. This file format allows users to conveniently store and transfer decks. By using the import menu, users can upload their saved decks to resume working on them at another time or on another device. The introduction of the import and export functionality improves the reusability of decks and allows users to distribute their decks to others, and thus engage in collaborative editing.

7. Maintenance and future iterations

The ongoing development and maintenance process of the application includes addressing bugs and resolving inconsistencies related to the user interface elements. The current version of the application already encompasses the essential editing functions. Future increments will focus on optimising operations, improving response times, and providing a more convenient and intuitive user experience.

Potential enhancements include the implementation of a login system with a backend database. By transferring a significant portion of the functionality to the backend, resource limitations on client-side browsers can be alleviated, particularly when working with larger decks. An additional benefit of a secure and well-designed authentication system would be the ability for users to safely store their decks in the database instead of downloading them locally onto their devices, facilitating easy sharing with other users directly through the application. Moreover, a stronger server-side would support incorporating version tracking capabilities, thus offering users broader opportunities to revert decks to previous states, as well as enabling undo and redo functionality for changes made during editing sessions. Additionally, the option to mark and delete multiple cards simultaneously and the ability to download all cards as images compressed in a single zip file could further improve the efficiency and usability of the application.

By considering these potential future increments, the Card Deck Designer can continue to evolve, providing extended functionality, improved performance, and a more comprehensive user experience.

8. Conclusion

In summary, through the design and testing of the YETI and Blue Yeti decks, it has been established that the Card Deck Designer provides all the essential tools for designing thematic card decks for didactic games. By offering LaTeX support, a user-friendly interface, customisation options, and efficient export capabilities, the Card Deck Designer has proven its functionality, versatility, and value in simplifying and enhancing the development process of didactic card games.

Furthermore, it is important to highlight that while our research primarily focused on the use of the Card Deck Designer in the field of mathematics, the versatility of the application extends beyond this domain. Its ability to incorporate scientific notations opens the door for various disciplines to benefit from its features, including physics, chemistry, and other subjects that utilise scientific formulas and notations. By offering a flexible and user-friendly platform for designing and customising card decks, the Card Deck Designer has the potential to support educators and game designers in a range of scientific disciplines, promoting interactive and engaging learning experiences across diverse subject areas. Therefore, the Card Deck Designer has the potential to further revolutionise the creation of educational card games and help provide new, immersive ways of learning for students worldwide.

To ensure continuous improvement of the Card Deck Designer, engaging educators and potential end users in the development is crucial. Their insights and ideas can largely contribute to future iterations of the application. Surveys, feedback sessions, and collaboration will uncover their needs and preferences, as well as guide the refinement of the application, creating an effective tool for educators and game designers to use.

9. Acknowledgements

Supported by the ÚNKP-22-1 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

Új Nemzeti Kiválóság Program MINISTRY OF CULTURE AND INNOVATION

References

[1] Hernández de Menéndez, M., Escobar Diaz, C. A., Morales-Menendez, R. (2020). Educational experiences with Generation Z. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 14, pp. 847–859. **https://doi.org/10.1007/s12008-020-00674-9**

[2] Qian, M., Clark, K. R. (2016). Game-based Learning and 21st century skills: A review of recent research. *Computers in Human Behavior*, 63, pp. 50–58. **https://doi.org/10.1016/j.chb.2016.05.023**

[3] Santos, A. (2017). Board games as part of effective game-based learning strategies. In: Spector, M., Lockee, B., Childress, M. (eds.). *Learning, Design, and Technology*. Cham: Springer, **https://doi.org/10.1007/978-3-319-17727-4_142-1**

[4] Moore, K., Jones, C., Frazier, R. (2017). Engineering education For Generation Z. *American Journal of Engineering Education (AJEE)*, 8, pp. 111–126. **https://doi.org/10.19030/ajee.v8i2.10067**

[5] Isaacs, A., Scott, S., Nisly, S. (2020). Move out of Z way Millennials. *Currents in Pharmacy Teaching and Learning*, 12, pp. 1387–1389. **https://doi.org/10.1016/j.cptl.2020.07.002**

[6] Árvai-Homolya Sz., Kőrei A., Lengyelné Szilágyi Sz. (2019). Élménypedagógiai játékok konstruálása tanulást támogató matematikai nyári tábor délutáni foglalkozásaihoz. *Matematikát, Fizikát és Informatikát Oktatók 43. Országos Konferenciájának tanulmánykötete*. DUE Press, 70–80. **https://www.uniduna.hu/images/MAFIOK_2019_tanulmnyktet_final_02.pdf**

[7]

[8] Lengyelné Szilágyi, Sz., Körei, A., Árvai-Homolya, Sz. (2019). Brainbox – Függvények. *GRADUS*, 6 (4), pp. 38−47. **https://gradus.kefo.hu/archive/2019-4/2019_4_CSC_005_Lengyelne.pdf**

[9] Szilágyi, Sz., Körei, A. (2022). Using a Math Card Game in several ways for teaching the concept of limit. *Mobility for Smart Cities and Regional Development – Challenges for Higher Education: Proceedings of the 24th International Conference on Interactive Collaborative Learning, ICL 2021*. Springer International Publishing, 2022/389, pp. 865–887. **https://doi.org/10.1007/978-3-030-93904-5_85**

[10] Dudás M., Lengyelné Szilágyi Sz., Piller I. (2019). Az *Ékkővadászok* elnevezésű matematikai készségfejlesztő kártyajátékok létrehozását támogató alkalmazás bemutatása. *GRADUS*, 6 (4), pp. 17–27. **http://real.mtak.hu/105946/**

[11] Szilágyi, Sz., Körei, A., Török, Zs., Vaičiulytė, I. (2023). Introducing the Card Editing Application of the LimStorm didactic game. *Professional studies: theory and practice*, 2023/12 (27), pp. 31–41. **https://doi.org/10.56131/pstp.2023.27.1.119**

[12] Lamport, L. (1994). *LATEX: A Document Preparation System: User's Guide and Reference Manual*. Addison-Wesley,New York, p. 291.

[13] Merriam, A. (2023). *LaTeX: A Gift to Scientific Publishing from Leslie Lamport*. https://www.cantorsparadise.com/latex-a-gift-to-scientific-publishing-from-leslie-lamport-a0b39492fc41.

[14] *LaTeX Documentation*. **https://www.latex-project.org/**

[15] Adenowo, A., Adenowo, B. (2013). Software engineering methodologies: A review of the waterfall model and object-oriented approach. *International Journal of Scientific and Engineering Research*, 4, pp. 427–434.

[16] Denning, P., Gunderson, C., Hayes-Roth, R. (2008). The profession of IT evolutionary system development. *Communications of the ACM*, 51 (12), pp. 29–31. **https://doi.org/10.1145/1409360.1409371**

[17] Kneuper, R. (2018). Software processes in the software product life cycle. *Software Processes and Life Cycle Models.* Cham: Springer, **https://doi.org/10.1007/978-3-319-98845-0_3**

[18] Ruparelia, N. B. (2010). Software development lifecycle models. *SIGSOFT Software Engineering Notes*, 35 (3), pp. 8–13. **https://doi.org/10.1145/1764810.1764814**

[19] Petersen, K., Wohlin, C. (2009). A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of Systems and Software*, 82, pp. 1479–1490. **https://doi.org/10.1016/j.jss.2009.03.036**

[20] Nielsen, J. (2010). *Websites Response Times. Nielsen Norman Group.* **https://www.nngroup.com/articles/website-response-times/**

[21] Palencsár, E., Szilágyi, Sz. (2023). Presenting the results of a three-pillar survey conducted among university students of IT to explore learning difficulties in the field of infinite series. *Multidisciplinary Sciences*, 13 (1), pp. 63–80. **https://doi.org/10.35925/j.multi.2023.1.7**