ΨΗΦΙΟΠΟΙΗΣΗ ΤΟΥ ΚΛΑΣΙΚΟΥ ΠΑΙΧΝΙΔΙΟΥ «PICTIONARY»

των

ΠΕΤΡΟΣ ΡΟΔΙΝΟΣ & ΤΣΟΥΝΤΑΝΗΣ ΕΛΕΥΘΕΡΙΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

υποβληθείσα σε μερική εκπλήρωση των απαιτήσεων για το πτυχίο
ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ



DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SCHOOL OF ENGINEERING

HELLENIC MEDITERRANEAN UNIVERSITY

2024

Εγκεκριμένη από:
Δρ. Βιδάκης Νικόλαος

DIGITIZATION OF THE CLASSIC GAME «PICTIONARY»

by

PETROS RODINOS & TSOUNTANIS ELEFTHERIOS

A THESIS

submitted in partial fulfillment of the requirements for the degree

BACHELOR OF INFORMATICS ENGINEERING



DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SCHOOL OF ENGINEERING

HELLENIC MEDITERRANEAN UNIVERSITY

2024

Approved by:

Dr. Vidakis Nikolaos

Περίληψη

Η επαρκής κατανόηση του λεξιλογίου από τον ανθρώπινο εγκέφαλο, ειδικά όταν πρόκειται για παιδιά, μπορεί να επιτευχθεί με αξιόλογα αποτελέσματα με τη χρήση διαδραστικών εκπαιδευτικών εργαλείων. Αυτή η προσέγγιση αποδεικνύεται πολύ πιο αποτελεσματική σε σύγκριση με τις παραδοσιακές μεθόδους διδασκαλίας που χρησιμοποιούνται συνήθως στη σημερινή εκπαίδευση η οποία ειναι αμετάβλητη εδώ και δεκαετίες.

Ειδικά όταν το διαδραστικό εκπαιδευτικό υλικό λαμβάνει τη μορφή ενός παιχνιδιού, όπως στην περίπτωση ενός παιχνιδιού "Pictionary," οι μαθητές έχουν την ευκαιρία να αναπτύξουν το λεξιλόγιό τους με διασκεδαστικό τρόπο. Σε κάθε γύρο, οι μαθητές πρέπει να αντιστοιχίσουν μια λέξη με βάση μια ζωγραφιά που τους παρουσιάζεται, εξασκώντας ταυτόχρονα την ορθογραφία τους. Αυτό τους ενθαρρύνει να αναζητήσουν νέες λέξεις, να ενισχύσουν τον λεξιλογικό τους θησαυρό, και να βελτιώσουν την ικανότητά τους να επικοινωνούν με ακρίβεια και ευκρίνεια.

Στόχος μας είναι να δημιουργήσουμε ένα φιλικό προς τον μαθητή περιβάλλον μέσω του παιχνιδιού "Pictionary," το οποίο θα προσφέρει μια ευχάριστη και εκπαιδευτική εμπειρία. Αυτό το παιχνίδι όχι μόνο θα ενισχύσει τις γνώσεις τους στον τομέα του λεξιλογίου, αλλά θα τους δώσει επίσης την ευκαιρία να αναπτύξουν κρίση, φαντασία και δημιουργικότητα, ενώ παράλληλα θα διασκεδάζουν και θα ανταγωνίζονται με τους συμμαθητές τους. Μέσα από αυτήν την προσέγγιση, προωθούμε την εκπαιδευτική καινοτομία και την επίτευξη πιο θετικών αποτελεσμάτων στην εκμάθηση του λεξιλογίο.

Abstract

The adequate comprehension of vocabulary by the human brain, especially when it comes to children, can be achieved with remarkable results using interactive educational tools. This approach proves to be much more effective compared to traditional teaching methods commonly used in today's educational practice, which have remained largely unchanged for decades.

Particularly when interactive educational materials take the form of a game, as in the case of a "Pictionary" game, students could develop their vocabulary in an enjoyable manner. In each round, students must match a word based on a presented drawing, simultaneously practicing their spelling. This encourages them to seek out new words, strengthen their vocabulary, and enhance their ability to communicate with accuracy and clarity. Our goal is to create a student-friendly environment through the "Pictionary" game, which will provide a fun and educational experience. This game will not only boost their vocabulary knowledge but also offer them the opportunity to develop judgment, imagination, and creativity while having fun and competing with their peers. Through this approach, we promote educational innovation and the achievement of more positive results in vocabulary learning.

Table of Contents

Abstract iv
Table of Contents v
List of Figures viii
List of Tables x
Chapter 1 - Introduction 1
1.1 1
1.2 Objective 2
1.3 Outline 3
Chapter 2 - Background Work 4
2.1 History Of Pictionary 4
2.2 ICT in education 6
2.3 Games in Education 7
2.4 Pictionary in vocabulary learning 8
Chapter 3 - Similar Work 9
3.1 Draw-and-Guess 9
3.2 Portalware 9
3.3 Pixelor 9
3.4 Fit to Draw 10
3.5 Sketch recognition using MGT 10
3.6 iNNk 11
3.7 Iconary 11
3.8 Learning English with AR Block Builder 11
3.10 Drawcto 12
Chapter 4 - Design 13
4.1 Problem Description 13
4.2 Software Analysis 14
4.2.1 Platform’s scenarios of use 14
4.2.2 Requirements 16
4.3 Architecture 18

4.3.1 Front End Structure 18
4.3.2 Web Access 19
4.3.3 Game Turn Mechanics 19
4.3.4 Pictionary Components Diagram 22
Chapter 5 - System Implementation 23
5.1 Technologies and Frameworks 23
JavaScript 23
Typescript 24
MongoDB 24
Node JS 25
Express JS 25
React 26
Vercel 26
Heroku 27
Cloudinary 27
ChatGpt 28
SCSS 28
5.2 Software Implementation 29
5.2.1 API part 29
5.2.1.4 Database Interaction with Mongoose 32
5.2.1.5 Client-Server Communication with Axios 33
5.2.2 Client 34
5.2.2.2 Form Management with React Hook Form 34
5.2.2.3 Reusability and Prop Creation 35
5.2.2.4 Cloudinary Integration for Image Upload 37
5.2.2.5 User Details Management 38
5.2.2.6 Room Creation 39
5.2.2.7 User Authentication State Management with Zustand 41
5.2.2.8 Word Generation with OpenAI 42
5.2.3 Hosting 44
5.2.3.1 Client Deployment With Vercel 44

5.2.3.2 Server-Side Deployment with Heroku 46
Chapter 6 - Result and Discussion 47
6.1.1 Login/Register 47
6.1.1.1 Register 48
6.1.2 Home page 49
6.1.3 Room Creation 51
6.1.4 Gameplay 53
6.2 Conclusion 56

List of Figures

Figure 2-1 . Pictionary Board Game 4
Figure 2-2. Pictionary: The Game of Quick Draw Gameplay MS-DOS 5
Figure 3-1 Fit to Draw gameplay 10
Figure 3-2 Process of AI integration with users drawing. 10
Figure 4-1. Front-End Architecture 18
Figure 4-2. Web Access Diagram 19
Figure 4-3. Round Mechanics 19
Figure 4-2. Web Access Diagram 20
Figure 4-3. Round Mechanics 21
Figure 4-4 Components 22
Figure 5-1. Port connection snippet 30
Figure 5-2. Socket connection snippet 30
Figure 5-3. Routing Snippet 31
Figure 5-4. DB Validation Snippet 32
Figure 5-5. Axios functions snippet 33
Figure 5-6. React Hook Snippet 34
Figure 5-7. Yup Validation Snippet 35
Figure 5-8. Prop Example Snippet 35
Figure 5-9. Prop User Snippet 36
Figure 5-10. Connecting to Cloudinary Server 37
Figure 5-11. Image upload snippet 37
Figure 5-12. Cloudinary Media Library 38
Figure 5-13. User Credentials after Registration 39
Figure 5-14. API Room Creation Snippet 39
Figure 5-15. Room Rules Snippet 40
Figure 5-16. Function Snippet 40
Figure 5-17. authStore Snippet 41
Figure 5-18. handleLogin Snippet 41
Figure 5-19. Connecting to OpenAI Snippet 42

Figure 5-20. AI Word Generation Command Snippet 43
Figure 5-21. Server Handling Generated Words Snippet 43
Figure 5-22. Versel Interface 44
Figure 5-23. Versel Analytics 45
Figure 5-24. Heroku Interface 46
Figure 6-1. Login demo 47
Figure 6-2. Register demo 48
Figure 6-3. Upload Avatar demo 48
Figure 6-4. Select Avatar demo 48
Figure 6-5. Home page demo 49
Figure 6-6. Settings tab top side 50
Figure 6-7 Customized settings tab bottom side 50
Figure 6-8 Language button 50
Figure 6-9. Room options 52
Figure 6-10. Join a room button 52
Figure 6-11. Choosing word 53
Figure 6-12. In Game Drawing tools 54
Figure 6-13 In Game Chat demo 54
Figure 6-14. Game Finished Tab 55
Figure 6-15. Updated home screen 55

List of Tables

Table 1. Teacher-User Stories 15
Table 2. Student User-stories 15
Table 3. Teacher user-stories requirements 16
Table 4. Student user-stories requirements 17

Chapter 1 - Introduction

1.1

In the ever-evolving landscape of education and technology, the convergence of gaming and vocabulary acquisition emerges as a captivating realm for exploration [1]. Rooted in creativity and communication, Pictionary, a timeless game, has not only captivated millions for its entertainment value but has also piqued interest in its potential educational impact [2]. This thesis embarks on a comprehensive journey, peeling back the layers of the Pictionary game, as it traces its historical roots, delves into its educational significance, and navigates through its technological evolution.

As we navigate the dynamic interplay between education and technology, the metamorphosis of the Pictionary game from its humble 1981 origins to its modern digital forms mirrors the transformative power of technology in reshaping the landscape of learning experiences. In the ever-expanding realm of digital gaming, the Pictionary game has not merely survived but has evolved as well, becoming a great choice for entertainment and leisure activities.

The digital evolution of Pictionary is intricately tied to technological advancements, fueled by the widespread availability of computing devices. No longer confined to the limitations of physical paper or tabletop setups, Pictionary has seamlessly transitioned into the digital sphere, capitalizing on the capabilities of modern devices to enhance the gaming experience.

Beyond its role as entertainment, digital Pictionary emerges as a tool with vast educational potential. This thesis seeks to unravel the multifaceted intersections of technology, creativity, and education within the Pictionary game. By examining its evolution, impact, and future possibilities, we aim to shed light on the transformative potential of this classic game in both educational and recreational spheres.

The digital adaptation of Pictionary not only preserves its traditional charm but also introduces innovative features. Real-time multiplayer modes, diverse drawing tools, and interactive elements augment the gaming experience, offering players a more immersive and engaging encounter. However, the journey does not end here; there is untapped potential for further exploration. Much like augmented reality enhances user interactions in various applications, the integration of novel interaction methods could revolutionize the Pictionary

experience. Gesture recognition or voice commands may introduce a new dimension of interactivity, propelling the game into new territories.

1.2 Objective

This thesis embarks on a transformative journey to reimagine the iconic Pictionary game, envisioning it as a dynamic and educational digital experience. Pictionary, traditionally a game of artistic expression and deduction, revolves around a participant sketching a word or phrase while others try to gues it based on visual clues. Our ambition is to recreate this classic by transforming it into a versatile tool for vocabulary enrichment.

Within the realm of digital games, a noticeable absence exists in platforms explicitly designed to support the vocabulary-building needs of students. Our vision is rooted in the belief that learning can be both effective and enjoyable. The absence of engaging educational games prompts to the creation of a Pictionary game that not only retains the joy of gameplay but also will serve as an valuable resource for vocabulary development.

Two key objectives on our game creation are flexibility and personalization. By incorporating user-friendly features such as personalized profiles, customizable interfaces, and interactive leaderboards, we aim to empower users to tailor their learning experiences. Students will have the ability to select word categories aligned with their interests and set difficulty levels that match their proficiency, creating a tailored educational journey.

Moreover, the integration of interactive leaderboards will add a competitive yet collaborative dimension to the game. We aim to foster a sense of community among players, turning vocabulary enhancement into a shared endeavor. This strategic blending of gaming dynamics and educational content will position our Pictionary game as a unique and promising solution in the modern educational landscape. In the vast and evolving landscape of learning tools, our Pictionary game will represent a step towards redefining how students engage with and benefit from educational games. Through this thesis, we aim to delve into the conceptualization and design principles that will found this educational game.

1.3 Outline

This thesis aims to contribute to the intersection of education and gaming, with a specific focus on vocabulary learning. Chapter 2 delves into the extensive history of Pictionary, tracing its evolution from a mere board game to its current state with limitless variations. This chapter highlights the pivotal role of Pictionary in vocabulary learning based on various research studies conducted on students. In Chapter 3, a examination of related works is conducted, exploring augmented reality (AR) implementations, artificial intelligence (AI) integrations, and advanced drawing programs. The analysis covers innovation gaps that previously existed in aligning technology with painting. The design of the game takes center stage in Chapter 4, providing an analysis of existing limitations in Pictionary when used for learning or teaching vocabulary. This chapter includes a software analysis on its key mechanics and presents the game's architecture through user stories, flowcharts, and component diagrams. Chapter 5 delves into the technologies employed to create the game, encompassing the frontend, backend structure, app hosting, and offering code snippets for key components to be established. In the final Chapter 6, the demonstration unfolds, offering readers a showcase of the game in action. This demonstration is crafted to emulate the experience of a user engaging with the app for the first time. Readers are guided through the user experience, providing insights into how the game integrates with the learning process and what it offers to users. The chapter concludes with a summary, bringing together key findings and insights gained from the demonstration. In chapter 7 we talk for our future visions on the game and potential upgrades .

Chapter 2 - Background Work

2.1 History Of Pictionary

The inception of the Pictionary game, as we recognize it today, can be traced back to its original creation, as can been seen in Figure 2-1. by a 20-year-old waiter named Rob Angel and his roommates in 1981[3]. This groundbreaking game was officially published in 1985 [4] under the banner of Angel Games. It drew inspiration from the widely popular Trivial Pursuit [5], a game that emerged in 1979 and gained prominence with its official publication in 1981. In the first year of its publication, Angel and his team managed to sell 6000 copies of that game for 35 dollars each.

In 1986, the licensing of Pictionary took a significant turn as Angel Games entered a joint venture with The Games Gang and Western Publishing. This collaborative effort led to the game's wider distribution and availability. In 1994 Hasbro assumed publishing responsibilities after acquiring the games business of Western Publishing.

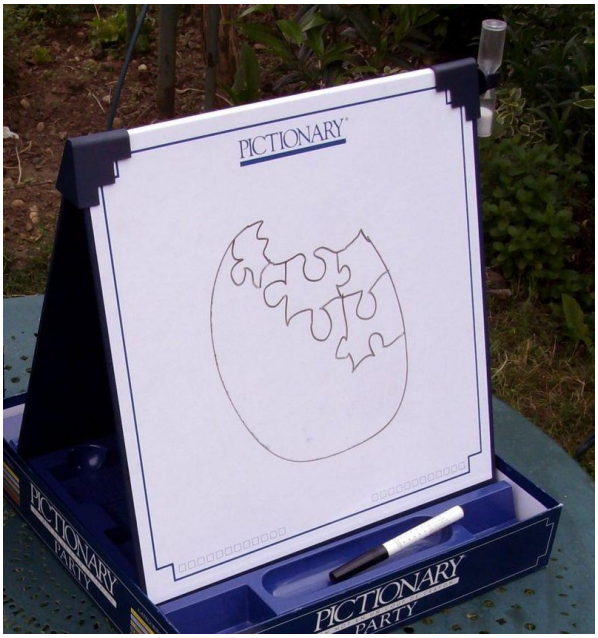


Figure 2-1 . Pictionary Board Game[6]

The transformative journey for the board game kept continuing in 2001 when Pictionary found a new home with the company Mattel, a game company colossus, marking a significant milestone

in its global reach. By that time, Pictionary had become a global phenomenon, being present in 60 countries and available in 45 languages. In the United States alone, there were 11 different versions of the game, contributing to a staggering 32,000,000 games sold worldwide [7].

The digital era embraced Pictionary in 1989 for MS-DOS and 1990 for Nintendo NES, marking its transition into the virtual realm as "Pictionary: The Game of Quick Draw." , as can been seen in Figure 2-2. This adaptation allowed players to engage in classic drawing and guessing game play either against other players or challenging computer-controlled opponents [8]. The digital format expanded the accessibility and reach of Pictionary, paving the way for a new era in its rich history.



Figure 2-2. Pictionary: The Game of Quick Draw Gameplay MS-DOS[9]

In September 2023 Mattel revolutionizes Pictionary with Pictionary Vs. AI, the first board game integrating AI technology. Player sketch, and AI guesses, introducing new challenges. Points earned by AI guesses and predicting accuracy. Featuring fun challenge rounds, the game utilizes Google's Quick, Draw! [10]. A modern twist to the classic since 1985[11].

2.2 ICT in education

Technology provides tools and functionalities that can enhance the caliber of teaching, albeit necessitating educators to incorporate fresh material into their lessons or pivot the instructional emphasis. [12]. Nonetheless, the adoption of technology and efficient teaching methodologies by educators hinges not solely on their individual experiences, perceptions, and beliefs. [13][14], but also it depends on the school's context they teach in [15][16] and the wider educational environment that could consider the innovation potential of a country and its economic growth [17].

According to Hatlevik and Hatlevik [18] collaborative efforts among colleagues promote the empirical utilization of digital technologies within classroom settings. Collaboration among teachers may play a key role in their ongoing professional development efforts within a school. To boost technology integration, educators must stay updated on new advancements and teaching methods. Effective school leadership can serve as a valuable means of supporting teachers in this endeavor. [19]. Clearly, ICT resources and technical support are necessary conditions for teaching with technology [20]. Additionally, teachers may find themselves in need of instructional support beyond the scope of traditional professional development programs. [16]. This assistance can be facilitated through teacher collaboration, professional development opportunities, or effective school leadership.

Incorporating technology into educational settings can significantly increase student engagement. Today's children are very familiar with digital devices, and using these tools for educational purposes can greatly capture their attention and improve their engagement. By integrating technology into learning, educators can offer students an immersive learning experience that keeps them focused and interested in the material. Furthermore, incorporating technology into the curriculum can also facilitate differentiated learning, allowing teachers to tailor instruction to meet the diverse needs of students. The use of advanced technological equipment, such as projectors and computers, in the classroom can make learning more interesting and enjoyable. Creating classroom activities that utilize technological tools, encourage collaborative work, and involve oral presentations can make the learning process more interactive and captivating. Additionally, technology allows for a variety of participation methods, not just through speaking [21][22][23].

2.3 Games in Education

According to Donmus [24], The concept of learning vocabulary through games has garnered significant attention, with educational games increasingly valued in language education [25]. These games are seen as valuable because they make language learning enjoyable and engaging. So, the learning model is needed for the students as the active learning. That can foster students' curiosity, particular in English. Then, Ahmad Rosidi, Eva Falianti [26] emphasized that vocabulary constitutes a crucial aspect of language education. Mastering vocabulary is essential as it enables individuals to effectively communicate through speaking, listening, and writing.

Techniques such as using games to teach and practice vocabulary are not commonly used. They are only used by both teachers and students for a time or occasion that can be powerless and useless. However, the strategic integration of game-based learning activities can transform these occasional practices into a powerful method of engaging students actively and effectively. Learning vocabulary through games offers numerous benefits. Prabha and Abdul Aziz [27] emphasized the importance of games providing learners with an enjoyable and relaxed learning experience. By engaging in games, students can practice using language in a stress-free manner after acquiring new vocabulary. In this context, students prioritize conveying their message over focusing on linguistic forms, allowing them to freely express themselves while maintaining the game's theme. This approach can alleviate students' fear of public assessment or negative evaluation, fostering a friendly environment that encourages learning and reduces anxiety [28].

Also, vocabulary knowledge can help improve the social potential of the student and helps in improving the communicative skills of the student. This enhancement in communicative skills is vital not only in academic settings but also in real-world interactions, where the practical use of language plays a key role in social integration and personal development. To foster students' vocabulary acquisition, diverse methods and approaches prove beneficial, as there exists no definitive blueprint for learning vocabulary. Each student possesses unique strategies for expanding their lexical repertoire. Research indicates that employing games as a tool for acquiring new vocabulary enhances retention and injects an element of enjoyment and motivation into language learning endeavors. [29][30][31]. Also, Murray and Ian, [32] accepted that engaging students in activities such as using games allows learners to recall new vocabulary more quickly. Game-based education helps the learning process to be fearless and meaningful.

2.4 Pictionary in vocabulary learning

Engaging in the Pictionary game offers several benefits. It serves as an effective educational tool for enhancing creative thinking skills and communication among students. Particularly beneficial for visual learners, Pictionary can help reinforce concepts across various subjects. Moreover, it facilitates the generation and reinforcement of a diverse range of facts, figures, and concepts. Ultimately, playing Pictionary fosters students' creativity and their ability to effectively communicate with their peers [33].That is possible only for a student who supports his/her classmates while they are in the proximal growth zone (students in a group have knowledge about the expected context, but they don’t know the actual word) [34]. Therefore, teachers need strategies, skills, or methods to positively influence and improve vocabulary learning. Iswandari [35] claims that teachers should be encouraged to employ games in the classroom to help students master new languages. Ensuring students' enjoyment while learning can be challenging. However, one effective approach is to incorporate games into English language teaching, providing children with a fun and engaging learning experience.

This game has been used in previous research analyzing its effectiveness to improve student’s vocabulary. This research is supported by Bunau [36] who says that Pictionary game is effective in teaching vocabulary. Learning vocabulary through games is highly enjoyable and has gained considerable popularity among both teachers and students. Sowell [37] believed that because they support making language education enjoyable, the importance of educational games has increased in language education. He observed that selecting games as a teaching medium can not only entertain but also educate, thereby enhancing the classroom atmosphere. For instance, using the game Pictionary can motivate previously disengaged students to participate actively in learning. Pictionary also aids students in comprehending words and their usage in English, facilitating easier memorization of vocabulary through practice in word generation and guessing based on visual cues.

Chapter 3 - Similar Work

By harnessing the synergy between the timeless joy of drawing and the ever-evolving landscape of technology, a new dimension of creativity, learning and enjoyment can be unlocked. Numerous studies have been conducted to explore this exciting intersection, shedding light on the possibilities and advantages it offers.

3.1 Draw-and-Guess

In the realm of drawing-based image retrieval games, several innovative approaches have been developed to engage users in generating useful data through gameplay. A noticeable one is Draw-and-Guess [38]. Which leverages a drawing-based image retrieval paradigm within the gameplay. In ‘Draw-and-Guess’, players take on the roles of both ‘painters’ and ‘guessers’ to draw and identify target images, respectively. This approach aligns with the objective of collecting valuable data for the field of drawing-based image retrieval.

3.2 Portalware

The authors of this study conducted a 14-week exploration of free-hand augmented reality (AR) sketching, commencing with a single-display smartphone configuration and identifying the limitations it posed. Subsequently, they introduced a dual-display smartphone-wearable setup called “Portalware”, a fingertip-mounted display, which enabled more fluid sketching. Their research yielded valuable insights into challenges and improvements for free-hand AR sketching, and they shared these findings through a detailed description of the “Portalware” system. The work also delves into the broader context of sketching in AR and VR, emphasizing the potential of smartphones and wearables for mobile AR sketching and highlighting the significance of wearables in enhancing the interaction experience and sketch quality.[39]

3.3 Pixelor

Of course, there isn't just one way to leverage Neural Networks in a game like Pictionary. One noteworthy example is 'Pixelor,' a Pictionary-inspired game where each player becomes a sketch artist. The game's premise involves everyone drawing while the Neural Network attempts to guess the word in real-time.The player whose drawing is guessed first emerges as the winner[40]

3.4 Fit to Draw

"Fit to Draw" is a distinctive multiplayer Pictionary-style game where your creativity is unleashed through physical movement alone. The game leverages GPS tracking to transform your real-world journeys into artistic expressions, using your continuous location as the brushstroke to create lines on the canvas of the game world, as can been seen in Figure 3-1.[41]

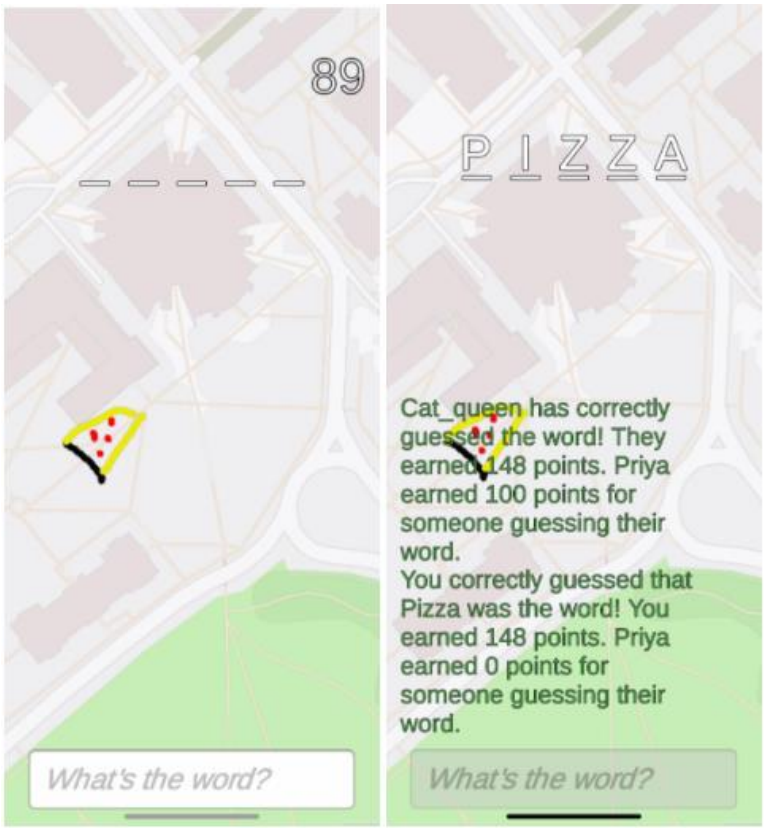


Figure 3-1 Fit to Draw gameplay[42]

3.5 Sketch recognition using MGT

In the context of sketch recognition and the representation of free-hand sketches, a method was introduced that involves representing sketches as multiple sparsely connected and applying a graph neural network known as the multigraph transformer (MGT) [43] to learn sketch representations. This approach effectively captures both global and local geometric stroke structures, as well as temporal information. This method demonstrates significant promise in the field of sketch recognition, offering an alternative to existing methods that rely on convolutional neural networks (CNNs) [44] and recurrent neural networks (RNNs) [45].

3.6 iNNk

“iNNk” is a drawing-based game featuring two teams: the Neural Network (NN) and the team of human players. In each round, one player assumes the role of the sketcher, while the others become guessers. Notably, the NN consistently acts as a guesser throughout the entire game. The primary objective of the game is to emerge victorious against the NN. The game's challenge lies in the fact that the NN continuously refines its abilities with each round and game, learning from the players' actions. Consequently, players must strategically outmaneuver the NN to prompt it to make incorrect word guesses [46].

3.7 Iconary

To enable AI to comprehend both metaphorical and literal human communication, developers designed “Iconary”, a game reminiscent of Pictionary, with a unique twist. In Iconary, players craft "drawings" by selecting and combining pre-existing images. While engaging in the game, AI collects and analyzes data from participants' attempts to predict the phrases being conveyed.

The primary aim of this game is to accumulate a vast repository of data, equipping AI with the capability to better grasp and interpret the intricacies of human language and communication.[47] While enjoying a game of Pictionary there is a chance that someone might cheat by writing the answer in order to get free points on a word that is hard to depict. In order to avoid that a team of developers created a deep neural atypical content detection network tool called “DrawMon” witch detects anomalies like text, numbers, question marks, arrows, circles and symbols on the canvas and registers them. And its purpose is to find those anomalies in order the gameplay to be fair for all the players.[48]

3.8 Learning English with AR Block Builder

In the realm of vocabulary-based learning games, noteworthy contributions have been made by integrating Augmented Reality (AR) technologies. An exemplary work explores the fusion of mobile gaming and AR to invigorate English vocabulary learning. This game transforms the conventional vocabulary learning process into a dynamic, multiplayer word-guessing experience. Players construct virtual figures for English words using an AR block builder, challenging others to decipher the words from these imaginative representations. The innovative approach not only facilitates vocabulary acquisition but also stimulates creativity. [49]

3.10 Drawcto

Inspired by the nature of non-representational art, developers have crafted “Drawcto”, a web-based, multi-agent AI drawing application. This innovative tool not only creates non-representational art but also engages in discussions about its specific features in collaboration with human artists. What makes “Drawcto” truly unique is its ability to generate non-representational art in the distinct styles of iconic artists such as Kandinsky, Picasso, and Monet. This achievement is made possible by harnessing the capabilities of machine learning and AI, along with the insights of Perceptual Organization Theory. Through this dynamic partnership, changes are integrated into the human artist's original sketch witch then gets its style transformed in to a drawing, ultimately resulting in a captivating non-representational masterpiece jointly crafted by both human and AI creativity , as can been seen in Figure 3-2 [50].

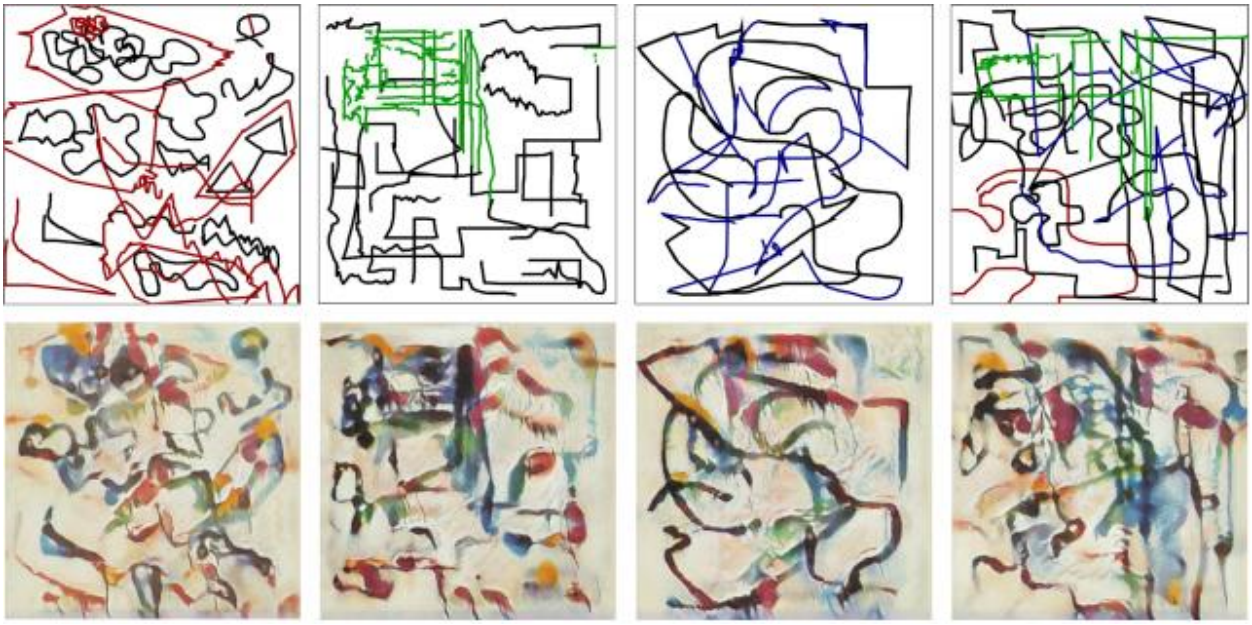


Figure 3-2 Process of AI integration with users drawing [51].

Chapter 4 - Design

This chapter delves into the intricacies of the toolset's software design. It includes an exploration of the toolset's requirements, the essential steps to fulfill these requirements, and an in-depth overview of the overall architecture, spotlighting its key components and functions. We will also delve into user stories, providing insights into the desirable in-game features aimed at enhancing students' vocabulary skills. Furthermore, the chapter analyzes the structural framework and design decisions incorporated into the proposed toolset to offer a comprehensive understanding of its development.

4.1 Problem Description

In contrast to many existing Pictionary games, customization and personalization options are often limited, leaving players without custom-made profiles and unaware of their in-game progress. While this might be acceptable for a simple Pictionary game, it becomes a crucial concern for a student-directed experience. Most games lack the flexibility needed to tailor gameplay to individual preferences, such as customizing characters or adopting alternative drawing methods beyond the classic mouse interaction.

Moreover, the absence of varied difficulty levels within the stagnant word pool and the inability to choose thematic word categories restrict the diversity of gameplay experiences. This deficiency in features becomes more pronounced when considering the unique needs of students trying to enhance their vocabulary skills. Additionally, fostering a sense of friendly competition is not actively encouraged in many existing games, which could be highly beneficial for students striving to improve.

The lack of customization options and features accommodating different learning preferences underscores a distinctive opportunity for the development of a Pictionary game explicitly designed for students. Standard words in Pictionary games, coupled with the absence of difficulty level options, further emphasize the need for a more tailored and student-centric approach in the realm of vocabulary enhancement games.

4.2 Software Analysis

In this chapter, we will explain our thinking behind the development of our vocabulary-enhancing Pictionary game. We will analyze the establishment of the game's objectives and primary functions, breaking down these goals into specific tasks to make easier to understand the implementation process. Our journey continues as we focus on the game's requirements, prioritizing its educational value and user engagement to ensure a smooth experience.

Furthermore, we delve into the intricacies of the game's design choices, carefully examining the visual elements and interactive features that have been thoughtfully considered. Alongside this, we present visual representations of the game's user interface and gameplay mechanics, offering readers a immersive view of the Pictionary game's design.

At its core, the Pictionary game aspires to redefine how students enrich their vocabulary through the compelling realm of visual learning. Within this interactive experience, players are challenged to interpret and identify various words through engaging images, fostering a dynamic connection between visuals and vocabulary. As a purpose-built educational tool, this game caters to students of all ages, providing a interactive avenue to expand their word knowledge.

Designed with versatility in mind, the target audience spans across students from diverse educational backgrounds, all seeking an enjoyable and effective means of enhancing their vocabulary through the immersive power of visual representation.

4.2.1 Platform's scenarios of use

To gain a comprehensive understanding of the Pictionary game's objectives and to structure the tasks necessary for its realization, we've adopted a similar approach of breaking down our development process through user stories. These personas encompass teachers and students, each with specific expectations and needs. In the domain of our vocabulary-enhancing Pictionary game, user stories serve to convey the value that each feature brings and offers insights into why users desire these specific functionalities. Moreover, to deem a user story as "complete," it must adhere to one or more acceptance criteria. These criteria lay out the precise requirements that must be fulfilled to ensure the successful implementation of each user story, thereby enhancing the educational value and user engagement within our vocabulary-enhancing Pictionary game.

Table 1. Teacher-User Stories

Teacher
As a teacher, I want to be able to split students into specific groups based on their vocabulary skills.
As a teacher, I want to encourage friendly competition among my students.
As a teacher, I would like to be able to customize the game's word lists to align with my curriculum and learning objectives.

Table 1 lists the teacher user-stories. These 3 user stories are essential for a classroom to run smoothly. They ensure fair competition among students while customizing the game to meet their specific educational needs and aligning with the subject of the lesson.

Table 2. Student User-stories

Student
As a student, I want the option to customize the game's visuals, such as avatars and themes, to make the learning experience more enjoyable.
As a student I want the game to has alternative and interactive ways of playing the game so it becomes more fun.
As a student, I want to be able to select my preferred difficulty level so that I can challenge myself at the right vocabulary level.
As a student, I want the game to has a scoring system that tracks my progress and improvements over time .
As a student, I want the game to provide a diverse range of words, so I can learn a wide variety of vocabulary.

Table 2 lists the student user-stories. These 5 user stories describe the features students want in the game. They want the game to be fun by allowing customization and offering different ways to play. Additionally, it should be an effective educational tool that helps improve their vocabulary and allows them to progress at their own skill level accordingly.

4.2.2 Requirements

The user stories outlined in section 4.2.1 can be subdivided into smaller tasks, which will help reveal the essential functionalities and specifications of the toolset. This approach clarifies what must be developed and the way it should be executed to attain the desired functionality.

In Table 3 the 3 user-stories about the teacher have been broken into smaller tasks as written on the right, and on the left the processes that need to be done to fulfill that user-story. As demonstrated, the user stories have revealed distinct tasks, each corresponding to a specific method for implementing the toolset components, ensuring an effortless user experience for designers.

Table 3. Teacher user-stories requirements

User-Story	**Tasks**	**Acceptance Criteria**
As a teacher, I want to be able to split students into groups according with their skills.	Make games rooms not accessible by everyone. So that players can create personal rooms to play .	Assign a unique room number to each game room.
As a teacher, I want to encourage friendly competition among my students.	Create a ranking system. Based on the games played by a player . So that they can get rewarded and keep track of their progress.	Store player experience points in a database and show it in game.
As a teacher, I would like to be able to customize the game's word pool to align with my curriculum and learning objectives.	Let the user choose the game's. word category when creating the game room. Let user choose game difficulty when creating the game room.	Generate a word pool using the AI's output, taking into account the selected word theme and difficulty level.

In Table 4 the 5 user-stories about the student have been broken into smaller tasks as written on the right, and on the left the processes that need to be done to fulfill that user-story. As demonstrated, the user stories have revealed distinct tasks, each corresponding to a specific method for implementing the toolset components, ensuring an effortless user experience for designers.

Table 4. Student user-stories requirements

User-Story	**Tasks**	**Acceptance Criteria**
As a student, I want the option to customize the game's visuals, such as avatars and themes, to make the learning experience more enjoyable.	Apply buttons that change the game theme. Which will allow character customization.	Add a set of theme assets in the game and establish a function for altering them. Implement an image upload function.
As a student I want the game to has alternative and interactive ways of playing the game so it becomes more fun.	Add the option for free hand drawing in the game so that players can draw using their cameras and hands.	Develop a selectable button in the canvas tools that allows the drawer to sketch using just their camera and hands.
As a student, I want to be able to select my preferred difficulty level so that I can challenge myself at the right vocabulary level.	Add difficulty levels in the game creation that will change the word difficulty.	Create difficulty level buttons that increase or decrease the difficulty of the words to be guessed and drawn.
As a student, I want the game to has a scoring system that tracks my progress and improvements over time .	Keep track of the scores after the games are played. Create a leveling system for the players.	Record scores in a database. Implement an experience point system, allowing players to level up as they play.
As a student, I want the game to provide a diverse range of words, so I can learn a wide variety of vocabulary.	Make the word pool get dynamically generated every time a game session gets created .	Make the words AI generated so Each game, even when sharing the same theme and difficulty level, will feature a unique set of words .

4.3 Architecture

In the upcoming subchapters, we will see the key aspects of our Pictionary game. The first section shows the intricacies of the front-end structure, detailing how components work together. The following part navigates through the user onboarding experience, including authentication and avatar selection. Lastly, we'll uncover the game round mechanics, outlining the player selection process and turn transitions. Visual aids, including a component diagram, will provide a overview of the frontend architecture, showing the layers that define the gameplay experience .

4.3.1 Front End Structure

The React Router within our architecture employs a nested routing approach, crafting a fluid user experience by rendering the layout and requested pages as interconnected child routes. In this structure, each page within the platform is composed of a specialized set of components. These components shoulder the responsibility of rendering specific content and adeptly managing user interactions. Notably, these components possess the capability to influence the application state by leveraging the actions module. Furthermore, they can initiate API requests, thereby facilitating the retrieval or modification of data from external sources. This well-orchestrated system ensures a seamless and responsive journey for users as they navigate through various pages and interact with diverse content within our vocabulary-enhancing Pictionary game.

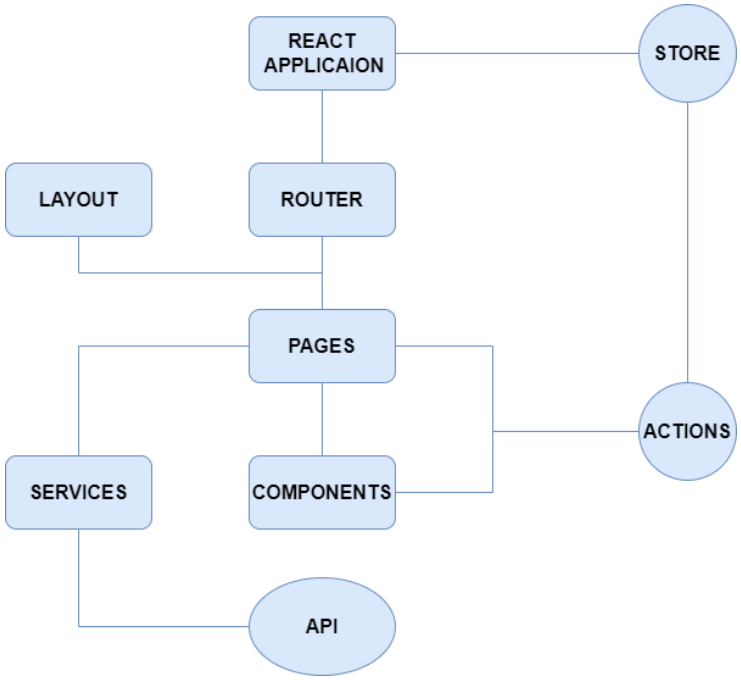


Figure 4-1. Front-End Architecture

4.3.2 Web Access

Upon entering the website, the system checks if the user is already logged in using Zustand to find a login token in the browser if created. If not, the player is prompted to log in by entering their account credentials or create a new account by providing necessary details and selecting an avatar image from the predefined pool or uploading a custom one. This straightforward process ensures a smooth onboarding experience, catering to both returning players and new users engaging with the game.

Note the diagram provided for the chapter 4.3.2 is the page 17 by the name :

Figure 4-2. Web Access Diagram

4.3.3 Game Turn Mechanics

Upon entering a game session, the system initiates the process of selecting a player who will take on the role of the drawer for the upcoming round, while the remaining players assume the role of guessers. The chosen drawer gains canvas access after selecting a word to illustrate from the provided options. If the drawer exceeds the allocated time for word selection, their turn is forfeited, and a guessing player takes over as the drawer. Once a word is chosen, the drawer utilizes canvas access to illustrate the word in order to find the solution. If the guessers successfully identify the word within the time frame, they earn points as a reward. When the timer concludes, the turn concludes, and another player is randomly selected as the new drawer, repeating the cycle until the game exhausts its designated rounds. This engaging and dynamic mechanism ensures an entertaining and competitive gameplay experience for all participants.

Note the diagram provided for the chapter 4.3.3 is the page 18 by the name :

Figure 4-5. Round Mechanics

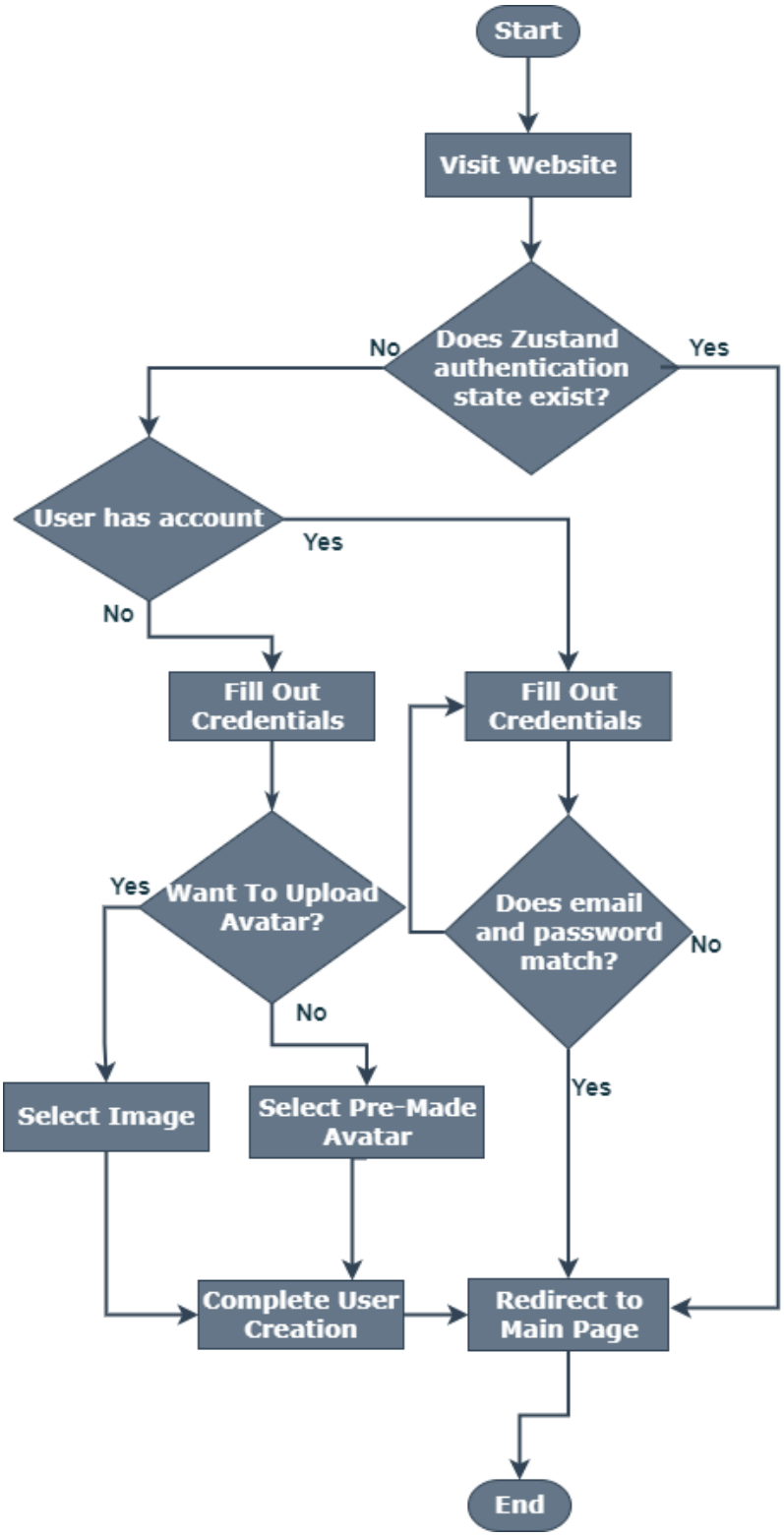


Figure 4-3. Web Access Diagram

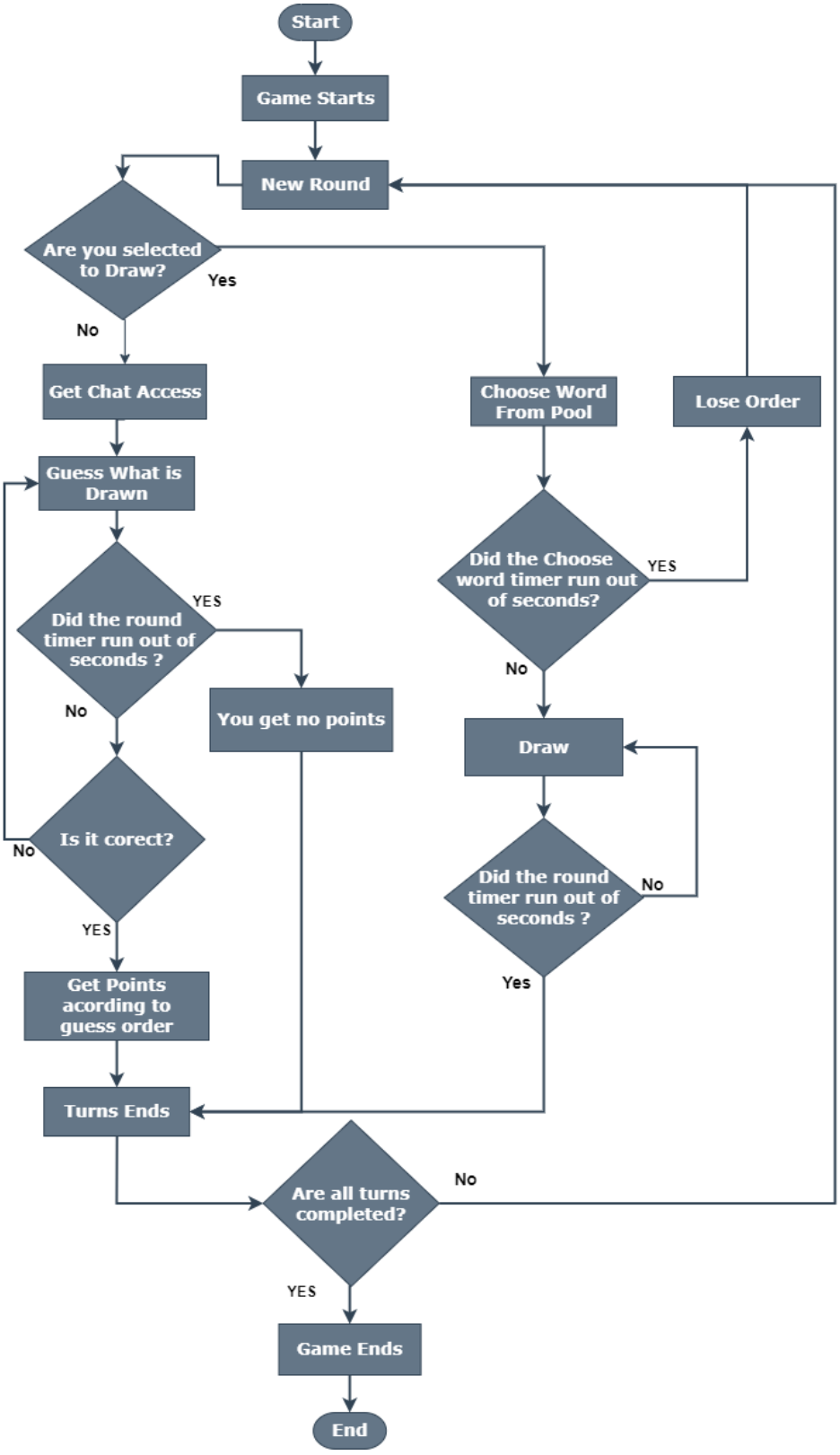


Figure 4-4. Round Mechanics

4.3.4 Pictionary Components Diagram

The component diagram visually delineates the architecture of the Pictionary game, offering a clear overview of its frontend structure. At the core is the React Application component, leading the entire frontend. This component connects with the Router, managing navigation intricacies and nested routing. The Zustand-powered Store component plays a pivotal role, ensuring connection state management shared across the application. Actions, as depicted in the diagram, act as the bridge, allowing modifications to the application state. The Layout component, linked to the Router, ensures a unified structure for each page, facilitating a consistent user interface. Pages, the dynamic facets of the game such as login, lobby, and drawing pages, emerge below the Layout. Nested within, the Components component embodies the modular nature of the pages, each consisting of distinct, purposeful elements. Completing the ensemble, the API component interfaces with Actions, embodying the communication between the frontend and backend, adeptly handling data retrieval and updates. This diagram encapsulates the interconnection of components that define the architecture of the Pictionary game, as can been seen in Figure4-6.

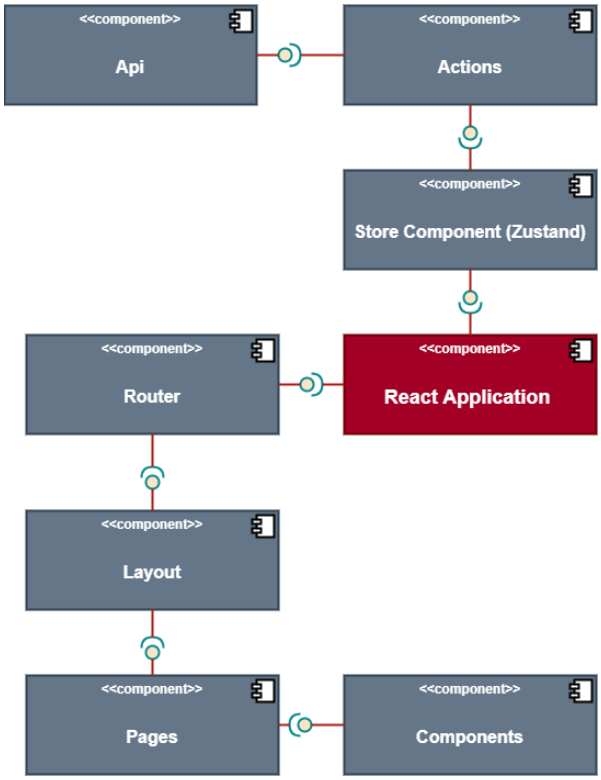


Figure 4-5 Components

Chapter 5 - System Implementation

In this section, we present a review of the technologies employed in the development of our application. We will delve into each technology's role and significance in the project, providing detailed paragraphs that illuminate the core components responsible for the application's functionality. This overview aims to offer readers a deeper understanding of the technological foundation that underpins our application, emphasizing the practical implementation of these tools and how they work together to deliver a seamless user experience.

5.1 Technologies and Frameworks

JavaScript

JavaScript, a versatile programming language primarily employed in web development, operates on the client-side, enhancing user interactivity and enabling dynamic modification of web page content without full-page reloads. It excels in responding to user actions like clicks and keystrokes and supports asynchronous communication with servers (AJAX) for real-time content updates [52]. To expedite development and create intricate web applications, developers frequently utilize JavaScript alongside libraries and frameworks like React, Angular, or Vue.js. Beyond web development, JavaScript has expanded its reach, entering server-side applications through platforms like Node.js. It has also found a place in mobile app development, utilizing tools like React Native, and desktop application development with Electron [53]. Despite its widespread use and adaptability in various fields, JavaScript has some challenges. Its single-threaded nature may result in performance bottlenecks, especially in resource-intensive tasks one notable challenge arises from its loosely-typed nature, where variables do not require explicit type declarations. This lack of strict typing can lead to unexpected behaviors and errors, JavaScript’s ubiquity and adaptability make it a fundamental language in the modern tech landscape, enabling the creation of a wide array of interactive web content and applications. While it comes with certain challenges, the language continues to evolve and remains a cornerstone in the toolkit of developers across diverse domains.

Typescript

TypeScript stands out as a powerful superset of JavaScript, offering a significant enhancement to the language with the incorporation of static typing and advanced tooling. By seamlessly integrating these features, TypeScript ensures compatibility with JavaScript while providing developers with a robust set of tools to elevate their coding experience.One of the key advantages of TypeScript lies in its ability to bolster code readability, making it easier for developers to understand and maintain their codebases. The introduction of static typing enables the early detection of type-related errors during the development phase, enhancing the overall safety and reliability of the code. This proactive approach to error identification contributes to a more streamlined and efficient development process.Notably, TypeScript has gained widespread acclaim in the realm of web development, where it proves instrumental in constructing modern, reliable, and scalable applications. Its utility becomes particularly evident in larger codebases and collaborative team environments, where the benefits of static typing and enhanced tooling shine through. The language's adaptability extends beyond the client-side, reaching into server-side development with Node.js integration, further expanding its versatility. [54].

MongoDB

MongoDB is a widely used NoSQL database known for its flexibility and schema-less data storage. Unlike traditional relational databases, MongoDB accommodates data without a fixed structure, making it adaptable to diverse data types and changing requirements. It utilizes a JSON-like document-oriented model, storing data in BSON format for efficiency and human readability. Scalability is a key strength of MongoDB, achieved through horizontal scaling via sharding and replication, making it suitable for managing large datasets and handling high traffic. The database features a robust query language for complex data operations, including geospatial and text search queries. MongoDB prioritizes high performance with options like indexing, caching, and in-memory storage. Designed for data integrity and availability, MongoDB employs features like replica sets. It has become a popular choice for web and mobile applications, content management systems, real-time analytics, IoT applications, and more. MongoDB's adaptability, scalability, and feature-rich design make it an indispensable tool for modern application development and database management. [55]

Node JS

Node.js is a powerful runtime environment that has revolutionized server-side application development. It is built on the V8 JavaScript engine and allows developers to use JavaScript on the server, creating a unified language and data format for both the front-end and back-end. What makes Node.js particularly significant is its non-blocking, event-driven, and asynchronous architecture, which enables it to handle a large number of concurrent connections efficiently, making it ideal for real-time applications and APIs [56]. This asynchronicity allows Node.js to execute multiple tasks simultaneously without waiting for each one to complete, leading to highly responsive and performant applications. Developers can write code that doesn't block the execution of other code, which is especially beneficial for handling I/O-bound operations, such as reading from databases or making network requests. Node.js has a robust package ecosystem, provided by npm, making it easy to access a wide variety of libraries and frameworks, further accelerating development. It is widely used in building web servers, microservices, and various network applications, often chosen for its speed, scalability, and developer-friendly features. Node.js has gained popularity in the developer community and is a valuable tool for creating high-performance, scalable, and real-time applications [57].

Express JS

Express.js, commonly referred to as Express, is a minimal and flexible web application framework for Node.js. It simplifies the process of building robust and scalable web applications and APIs. Express offers a lightweight set of features and middleware that allows developers to handle routing, request handling, and other common web application tasks with ease. It follows the principles of Node.js, which is known for its non-blocking and asynchronous I/O, making it a suitable choice for building high-performance applications [58]. Express simplifies the creation of routes, handling HTTP requests, and connecting to databases. It also provides a way to manage and customize various aspects of an application, such as middleware, authentication, and error handling. Due to its versatility, Express is often used in conjunction with different databases and frontend libraries like React or Angular, offering a complete solution for web application development. Its active community and numerous plugins and extensions have solidified Express as a top choice for developers looking to build efficient and scalable web applications or APIs [59].

React

React, also known as React.js or ReactJS, is an open-source JavaScript library developed and maintained by Facebook. It is widely used for building user interfaces for web applications. What makes React stand out is its component-based architecture, allowing developers to create encapsulated, reusable UI components [60]. These components are composed to build complex user interfaces, enabling a more organized and maintainable codebase. React employs a virtual DOM (Document Object Model) that efficiently updates only the parts of a webpage that have changed, reducing the need for full page reloads and enhancing performance. React's unidirectional data flow and declarative nature make it predictable and easier to debug. It is often used in conjunction with other technologies, like Redux or React Router, for state management and routing. React's extensive ecosystem, including tools like React Native for mobile app development, has made it a popular choice for building interactive and responsive web interfaces. Its flexibility, scalability, and active community support have established it as a fundamental library for modern web development.[61]

Vercel

Vercel is a cloud platform and a serverless deployment service designed to simplify web application deployment and hosting. It specializes in providing a seamless experience for front-end developers by focusing on the JAMstack architecture (JavaScript, APIs, and Markup). Vercel is especially known for its ease of use, speed, and automation, allowing developers to deploy websites and applications with minimal configuration. With its serverless functions, Vercel enables the execution of code on-demand, providing cost-efficiency and scalability as resources are allocated dynamically. It supports a wide range of programming languages, making it versatile for various project needs. Vercel's continuous deployment and integration capabilities, combined with its global content delivery network (CDN), ensure fast and reliable delivery of web content to end-users worldwide. Its developer-friendly features have made it a popular choice for hosting web applications and static websites, and it integrates seamlessly with various front-end frameworks and version control systems. Vercel's focus on simplicity and performance makes it a valuable platform for developers looking to streamline their deployment and hosting processes.[62][63]

Heroku

Heroku is a cloud platform that offers a convenient and flexible environment for developers to deploy, manage, and scale web applications with ease. It simplifies the deployment process by providing a platform-as-a-service (PaaS) model, taking care of infrastructure management, and allowing developers to focus on coding. Heroku supports various programming languages and frameworks, making it adaptable to a wide range of application types. What sets Heroku apart is its simplicity and the ability to scale applications effortlessly, both vertically and horizontally, ensuring that they can handle increased traffic and demand. It offers a rich ecosystem of add-ons and buildpacks, making it easy to enhance application functionality with databases, caching, and other services. Heroku's continuous integration and delivery (CI/CD) capabilities streamline the development workflow, and its containerization options provide additional flexibility. With a strong developer community and extensive documentation, Heroku is a popular choice for startups and established businesses seeking a platform that simplifies deployment and enables rapid scaling of web applications.[64][65]

Cloudinary

Cloudinary is a cloud-based media management platform that simplifies the process of storing, managing, and delivering images and videos for web and mobile applications. It offers developers a convenient way to optimize, transform, and serve media assets, reducing the complexity and time required for media management. Cloudinary's features include powerful image and video manipulation capabilities, such as resizing, cropping, and format conversion, all accessible through a straightforward API. This makes it an excellent choice for developers looking to deliver high-quality, optimized media content to end-users while maintaining a fast and responsive user experience. Furthermore, Cloudinary's ability to deliver media efficiently across global content delivery networks (CDNs) ensures that users worldwide receive content quickly. With a strong emphasis on developer-friendliness and a wide range of SDKs and integrations for popular web and mobile frameworks, Cloudinary has become a popular choice for media management, simplifying the process of handling media assets in modern applications.[66]

ChatGpt

ChatGPT is a language model created by OpenAI. It uses the GPT (Generative Pre-trained Transformer) architecture to generate human-like text based on the input it receives. Essentially, it's a tool for natural language understanding and generation. In our case it is used
for word generation in order to enhance the word pool of the game sesions .[67]

SCSS

SCSS, which stands for "Sassy CSS," is a scripting language that is a superset of CSS (Cascading Style Sheets). SCSS uses syntax similar to CSS but includes additional features and capabilities that make it more powerful and easier to work with. It is often referred to as a CSS preprocessor because it allows developers to write code in a more concise and maintainable way, and then it is compiled into standard CSS that web browsers can understand.[68]

Handtrack.js

Handtrack.js revolutionizes the way developers interact with hand pose detection within the web environment. This advanced library empowers developers to accurately identify various hand poses and facial expressions directly in the browser, utilizing a robust object detection neural network model. Optimized for JavaScript through TensorFlow.js[69], Handtrack.js brings the power of machine learning to the fingertips of web developers, offering the capability to detect six distinct hand poses—open, closed, pinch, point, face, pointtip, pinchtip—as well as faces, enabling a more interactive and engaging user experience.[70]

5.2 Software Implementation

In this chapter, this thesis unveils the essential code snippets needed for the functionality of the Pictionary game development. The presentation sequence mirrors the user's journey with the application, encompassing stages such as character creation, transitioning to the waiting room, engaging in gameplay, and acquiring experience points after participating in gaming sessions.

The chapter will be divided in 3 parts the first is the background which is the api implementation , the second part is the client implementation and the third is the hosting services used to make the game run on the web instead of locally.

5.2.1 API part

In the API implementation, we delve into the intricate collaboration of technologies aimed at crafting a robust server with real-time communication capabilities. Leveraging the MERN stack, Node.js, and the Express framework provide the backbone for server development, offering a scalable and efficient solution. The integration of Socket.io enhances the application's real-time features, enabling seamless communication between clients. Mongoose, as an ODM library, facilitates the interaction with the MongoDB database, ensuring secure and organized storage of user data. Additionally, user authentication is fortified using bcrypt for password hashing and JWT for secure token generation. The Axios library further bridges the gap between the client and server, streamlining HTTP requests for user authentication, registration, and data updates. Together, these technologies converge to create a dynamic and responsive API for our multiplayer Pictionary game.

5.2.1.1 Server Setup with Node.js and Express

In the process of creating a multiplayer game, the first step involves setting up a server. To streamline this process, Node.js is employed, and Express, a minimalist web application framework for Node.js, is used for easier server-side development. The server is responsible for handling client requests, managing connections, and facilitating communication between players, as can been seen in Figure 5-1.

```
const PORT: number = Number(process.env.PORT) || 3000;
mongoose.connect(process.env.MONGO_URI).then(() => {
  http.listen(PORT, () => {
    console.log(`Server running on Port: ${PORT}`);
  });
});

```

Figure 5-1. Port connection snippet

In the code snippet above, the server is configured to listen on a specified port (defaulting to 3000 if not provided through an environment variable). Additionally, a connection to a MongoDB database is established using Mongoose, a MongoDB object modeling tool designed to work in an asynchronous environment.

5.2.1.2 Real-time Communication with Socket.io

For a real-time multiplayer experience, Socket.io is integrated into the application. Socket.io enables bidirectional communication between the server and connected clients. This is crucial for instant updates and synchronization among players , as can been seen in Figure 5-2.

```
const socket = io(http, {
  cors: {

    origin: "*",
    methods: ["GET", "POST", "PUT", "DELETE"],
    "Access-Control-Allow-Origin": "*",
  },
});
```

Figure 5-2. Socket connection snippet

The code initializes a Socket.io instance, configuring it to accept connections from any origin (*) and specifying allowed HTTP methods. This enables seamless communication between the server and clients while handling cross-origin resource sharing (CORS) [71] efficiently.

5.2.1.3 Routing and User Authentication

In a multiplayer game, user authentication and routing play a crucial role in managing user data, securing sensitive endpoints, and providing a seamless gaming experience. Express.js is utilized for setting up routes, and middleware functions are implemented to ensure that only authenticated users can access certain endpoints, as can been seen in Figure 5-3.

```
5   const router = express.Router();
6
7   router.post("/user/register", usersController.register);
8
9   router.post("/user/login", usersController.login);
10
11  router.put("/user/:id", authMiddleware, usersController.updateUser);
12
13  router.get("/user/:id", authMiddleware, usersController.getUser);
14
15  router.get("/users", authMiddleware, usersController.getUsers);
```

Figure 5-3. Routing Snippet

This section has covered the routing aspect of the Pictionary game, focusing on user authentication and data management. By employing Express.js for routing and middleware functions for user authentication, the application ensures secure access to different endpoints.

- The /user/register and /user/login endpoints handle user registration and login, respectively. These routes interact with the usersController module, which contains the logic for user authentication and registration. Proper validation and error handling are essential here to ensure the security and reliability of the authentication process.
- Endpoints such as /user/:id, which allows users to retrieve their own information, and /users, which provides a list of all users, are crucial for managing user data. The authMiddleware function is used as middleware to ensure that only authenticated users can access these routes.
- The authMiddleware function is a middleware that checks the validity of authentication tokens. This function is employed as middleware for routes requiring authentication, such as updating user information and retrieving user details. It adds an additional layer of security by verifying the user's identity before allowing access to sensitive data.

5.2.1.4 Database Interaction with Mongoose

Effective interaction with the database in a multiplayer game is crucial for managing user data securely. Mongoose, an ODM (Object Data Modeling) library for MongoDB and Node.js, facilitates this interaction by providing a schema-based solution. The example below demonstrates the login functionality, utilizing Mongoose to query the database for user information, as can been seen in Figure 5-4.

```
64  export const login = async (req: Request, res: Response, next: NextFunction) => {
65    const { username, password: userPassword } = req.body;
66
67    try {
68      const user = await User.findOne({ username });
69
70      if (!user) {
71        return res.status(404).json({
72          message: "Invalid username/password combination",
73        });
74      }
75
76      const checkPassword = bcrypt.compareSync(userPassword, user.password);
77      if (!checkPassword) {
78        return res.status(404).json({
79          message: "Invalid username/password combination",
80        });
81      }
82
83      const loggedInUser = exclude(user.toObject(), "password");
84
85      const accessToken = await jwt.signAccessToken({
86        id: loggedInUser._id,
87        username: loggedInUser.username,
88      });
89
90      return res.status(200).json({
91        token: accessToken,
92        ...loggedInUser,
93      });
94    } catch (error) {
95      return res.status(500).json({
96        message: "An error occurred while trying to log in",
97      });
98    }
99  };
```

Figure 5-4. DB Validation Snippet

This code snippet demonstrates the process of authenticating a user during the login operation. The user's entered password is compared with the hashed password stored in the database. If the credentials are valid, an access token is generated using a JWT library, allowing the user to access protected routes.

Error handling is implemented to gracefully manage scenarios where the user is not found or where there are issues with password validation. The server responds with appropriate status codes and error messages to provide clear feedback to the client.

5.2.1.5 Client-Server Communication with Axios

In our Pictionary game, effective communication between the client and server is vital for transmitting user actions, game state updates, and other relevant data. Axios, a popular HTTP client library, simplifies this process by providing a straightforward API for making asynchronous HTTP requests, as can been seen in Figure 5-5.

```
12  export const loginUser = async (paylaod: UserLogin): Promise<any> => {
13    try {
14      const result = await axios.post(`${API_URL}user/login`, paylaod);
15      return result.data;
16    } catch (err: any) {
17      console.log("err", err);
18      throw err?.response?.data?.message;
19    }
20  };
21
22  export const registerUser = async (paylaod: NewUser): Promise<any> => {
23    try {
24      const result = await axios.post(`${API_URL}user/register`, paylaod);
25      return result.data;
26    } catch (err: any) {
27      console.log("err", err);
28      throw err?.response?.data?.message;
29    }
30  };
31
32  export const updateUser = async (payload: UserToUpdate): Promise<any> => {
33    try {
34      const result = await axios.put(`${API_URL}user/${payload.userId}`, payload, getConfig());
35      return result.data;
36    } catch (err: any) {
37      console.log("err", err);
38      throw err?.response?.data?.message;
39    }
40  };
```

Figure 5-5. Axios functions snippet

How Axios Functions work ?

They send a request to the server's endpoint with user credentials.

If successful they return the data received from the server.

If an error occurs, they log the error and throw a customized error message if available.

5.2.2 Client

At this chapter we will see the technologies used in the Client side of our Pictionary game, also known as front end. And we will understand how they help to accomplish a solid user interface.

5.2.2.2 Form Management with React Hook Form

In the client-side implementation, we use React Hook Form for the creation and management of forms within our multiplayer Pictionary game. React Hook Form is a versatile library that uses hooks to simplify form development. The snippet below showcases key functionalities provided by React Hook Form, such as form registration, submission handling, error tracking etc..as can been seen in Figure 5-6.

```
49  const {
50    register,
51    handleSubmit,
52    reset,
53    formState: { errors },
54    setValue,
55    getValues,
56  } = useForm<any>({
57    resolver: yupResolver(isEditing ? EditProfileValidationSchema : RegisterValidationSchema),
58    defaultValues: {
59      username: "",
60      password: "",
61      role: "",
62      age: "",
63      avatar: "",
64    },
65  });
```

Figure 5-6. React Hook Snippet

5.2.2.2.1 Yup for Validation

Yup is a JavaScript schema builder for value parsing and validation. In conjunction with React Hook Form, we utilize Yup for flexible form validation. Yup allows us to define clear validation schemas for our data, ensuring that user input adheres to specified criteria. The validation schema for user registration is exemplified below, as can been seen in Figure 5-7.

```
8   export const RegisterValidationSchema = yup.object({
9     username: yup.string().required("Username is required"),
10    password: yup.string().required("Password is required"),
11    role: yup.string().required("Role is required"),
12    age: yup.string().required("Age is required"),
13  });
```

Figure 5-7. Yup Validation Snippet

The React Hook Form instance is configured with the defined validation schema. The form's **handleSubmit** function is linked to the form submission logic, ensuring that data is validated before being processed. The error state, managed by React Hook Form, allows us to efficiently handle form errors and provide users with meaningful feedback.

5.2.2.3 Reusability and Prop Creation

React's emphasis on component-based architecture is exemplified through the creation and utilization of props. Props, short for properties, serve to pass data from parent components to their child components. This practice not only enhances the reusability of components but also contributes to the overall modularity and maintainability of the codebase, as can been seen in Figure 5-8.

```
7   interface ImageUploaderProps {
8     style?: React.CSSProperties;
9     className?: string;
10    onChange: (image: string) => void;
11    name?: string;
12    label?: string;
13    value?: string;
14  }
```

Figure 5-8. Prop Example Snippet

5.2.2.3.1 Benefits of Prop Creation

Reusability: By encapsulating functionality within a component and defining props, the component becomes reusable in various contexts. The ImageUploader component, for instance, can be employed with different labels, names, and initial values, enhancing code reusability.

Modularity: Props contribute to the modularity of the codebase. Components are designed to perform specific tasks, and their behavior can be easily adjusted by modifying the props they receive.

Customization: Props offer a way to customize the appearance and behavior of components. In the example, the style, className, and other props provide flexibility in tailoring the visual presentation of the ImageUploader component.

In the provided example, the ImageUploader component is utilized with specific props.

The value prop is set based on the state or data received from the parent component.

The onChange prop is assigned a callback function (handleAvatarChange) to handle changes to the image selection. Additional props like name and label are used to customize the component's behavior and appearance, as can been seen in Figure 5-9.

```
235          <ImagePicker
236            value={getValues("avatar") || values?.avatar}
237            onChange={handleAvatarChange}
238            name="avatar"
239            label={t("select-avatar")}
240          />
```

Figure 5-9. Prop User Snippet

React's power lies in its ability to enhance code reusability through encapsulation and the use of props. It promotes modularity by allowing components to perform specific tasks, easily adjustable through props. The customization offered by props, such as styling and behavior adjustments, adds flexibility to component design. React's simplicity in handling props makes it a powerful technology for building modular, reusable, and customizable user interfaces.

5.2.2.4 Cloudinary Integration for Image Upload

In the client-side implementation, Cloudinary emerges as a powerful asset for handling image uploads, storage, and retrieval. By seamlessly connecting to Cloudinary, our Pictionary game enhances user experience by efficiently managing user avatars. Below is an overview of the Cloudinary integration, as can been seen in Figure 5-10.

```
cloudinary.config({
  cloud_name: process.env.CLOUDINARY_CLOUD_NAME,
  api_key: process.env.CLOUDINARY_API_KEY,
  api_secret: process.env.CLOUDINARY_API_SECRET,
});
export { cloudinary };
```

Figure 5-10. Connecting to Cloudinary Server

The Cloudinary integration facilitates the upload of user avatars, contributing to a visually enriched profile experience. In the example below, the cloudinary.uploader.upload function is employed to upload an avatar image to the Cloudinary server. The folder option organizes the uploaded images into a designated "avatars" folder, as can been seen in Figure 5-11.

```
try {
  if (avatar.length <= 500) {
    avatarUrl = avatar;
  } else {
    const result = await cloudinary.uploader.upload(avatar, {
      folder: "avatars",
    });
    avatarUrl = result.url;
  }
```

Figure 5-11. Image upload snippet

5.2.2.4.1 Benefits of Cloudinary Integration

Efficient Image Handling: Cloudinary streamlines the process of image handling, offering a reliable and scalable solution for image uploads and storage.

Optimized User Experience: By offloading the responsibility of image storage to Cloudinary, the Pictionary game ensures a smooth and responsive user experience, especially when dealing with avatar images, as can been seen in Figure 5-12.

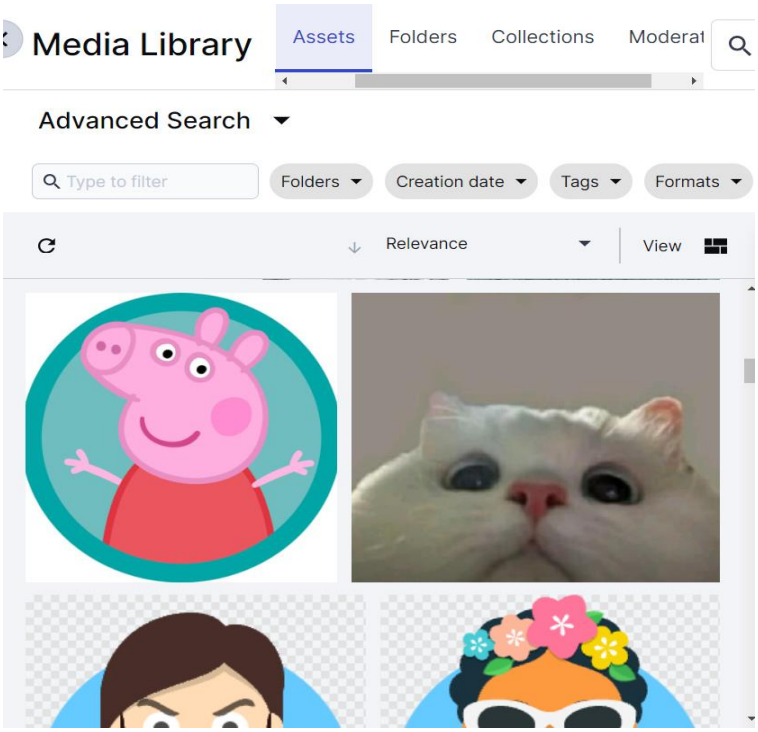


Figure 5-12. Cloudinary Media Library

5.2.2.5 User Details Management

Upon successful registration, the user details can be utilized to provide a personalized and informative profile section within the Pictionary game's interface. React components can be created to render the user details dynamically , as can been seen in Figure 5-13.



Figure 5-13. User Credentials after Registration

As the user progresses in the game, you might want to update and display real-time changes to the user details. For example, if the user gains experience points or levels up, these changes can be dynamically reflected inside of the game .

5.2.2.6 Room Creation

Room Creation when used begins on the server side, the create-room socket event triggers the creation of a game room. The server initializes the room with the specified settings, including the creator's user ID, and sets up essential room properties, such as players, drawings, status, round information, chat history, and timers, as can been seen in Figure 5-14.

```
47  socket.on("connection", (socket: any) => {
48    //creating room
49    socket.on("create-room", async (settings: Room) => {
50      rooms[settings.code] = {
51        ...settings,
52        players: [],
53        drawings: "",
54        status: Statuses.CREATED,
55        round: 1,
56        message: "",
57        lastWord: "",
58        chat: [],
59        usersFoundWordOrder: [],
60      };
61      timers[settings.code] = {
62        round: undefined,
63        choosingWord: undefined,
64      };
65    });
```

Figure 5-14. API Room Creation Snippet

To ensure a fair and enjoyable gaming experience, certain rules are applied during the room creation process. The client-side handleCreateRoom function enforces rules related to the completion of all required fields, prompting a warning if any are left empty. Additionally, on the server side, the room is initialized with default values and properties, setting the stage for a seamless multiplayer Pictionary game, as can been seen in Figure 5-15.

```
49  const handleCreateRoom = () => {
50    if (!settings.choosingWordTime || !settings.roundTime || !settings.maxPlayers)
51      return toast.warn(t("fill-out-all-fields"));
52
53    let gameSettings = {
54      ...settings,
55      maxPlayers: Number(settings.maxPlayers),
56      choosingWordTime: transformToMilliseconds(settings.choosingWordTime),
57      roundTime: transformToMilliseconds(settings.roundTime),
58    };
59    if (categories.includes(settings.category)) {
60      gameSettings.customWords = JSON.parse(words)[settings.category][settings.difficalty];
61    }
62    onCreate(gameSettings);
63  };
```

Figure 5-15. Room Rules Snippet

A multiplayer experience, with game rooms being the focal point where drawings and guesses unfold. The client-side implementation involves the initiation of a room creation process, allowing players to customize game settings. The handleCreateRoom function exemplifies this process, as can been seen in Figure 5-16.

```
55  const handleCreateRoom = (settings: GameSettings) => {
56    socket?.emit("create-room", {
57      ...settings,
58      creator: userId,
59    });
60    setSearchParams({
61      room: settings.code,
62    });
63  };
```

Figure 5-16. Function Snippet

5.2.2.7 User Authentication State Management with Zustand

In the client-side implementation of the Pictionary game, Zustand plays a pivotal role in managing the state related to user authentication. The **authStore** defines the authentication state, providing methods to log in, log out, and update the user profile. Additionally, Zustand integrates persistence, ensuring that user credentials persist even when the browser is refreshed or closed, as can been seen in Figure 5-17 & Figure 5-18.

```
export const authStore = create<AuthState>()(
  devtools(
    persist(
      (set) => ({
        ...initialStateValues,
        logOut: () => {
          set({
            ...initialStateValues,
          });
        },
        logIn: (payload: any) =>
          set({
            isLoggedIn: true,
            token: payload.token,
            userId: payload.userId,
            username: payload.username,
            level: payload.level,
            xp: payload.xp,
            avatar: payload?.avatar,
            role: payload.role,
            categories: payload.categories,
            words: payload.words,
          }),
        updateProfile: (payload: any) =>
          set((state) => ({
            ...state,
            ...payload,
          })),
      }),
      {
        name: "auth-pictionary",
      }
    )
  )
);
```

Figure 5-17. authStore Snippet

```
const handleLogin = async (values: UserLogin) => {
  loginMutation(
    {
      username: values.username,
      password: values.password,
    },
    {
      onSuccess: (data: any) => {
        if (data?.token) {
          logIn({
            userId: data._id,
            username: data.username,
            token: data.token,
            level: data.level,
            xp: data.xp,
            avatar: data.avatar,
            role: data.role,
            categories: data.categories,
            words: data.words,
          });
          navigate("/home");
        } else {
          toast.error(t("invalid-username"));
        }      You, 4 months ago • navigating to hom
      },
      onError: (error: any) => {
        if (error) {
          toast.error(error);
        } else {
          toast.error(t("could-not-login"));
        }
      },
    }
  );
};
```

Figure 5-18. handleLogin Snippet

The **handleLogin** function illustrates how Zustand is utilized to manage user authentication. Upon a successful login, the logIn method from the authStore is invoked, updating the authentication state with the user's information. The user is then navigated to the home page.

5.2.2.7.1 Benefits of Zustand for Authentication State

Simplicity and Intuitiveness: Zustand simplifies state management by providing a clean and intuitive API, making it easy to handle complex state logic, such as user authentication.

DevTools Integration: The use of devtools allows developers to inspect and debug the Zustand state effortlessly, enhancing the development process.

Persistence: Zustand's persist middleware ensures that user authentication state persists across browser sessions, enhancing the user experience.

5.2.2.8 Word Generation with OpenAI

During the gameplay , OpenAI GPT-3.5 is used to dynamically generate word pools for different categories and difficulty levels. The integration involves utilizing the openAi instance, specifying the prompt for word generation, and processing the response to obtain the desired words, as can been seen in Figure 5-19.



Figure 5-19. Connecting to OpenAI Snippet

As can been seen in Figure 5-20 the Prompt function constructs a prompt tailored for word generation in JSON format. The prompt strictly orders the AI to generate words based on :

- The word category given.
- The number of words given.
- The difficulty level given.
- The language given.

```
export const Prompt = (category: any, words: any, difficultyLevels: any) => {
  return `given the word: ${category},generate STRICTLY ${words} single objects for each category
   [${difficultyLevels}] related to this for an online drawing game in the format of
    {
    easy:[word1,word2,...${words}]
    medium:[word1,word2,...${words}],
    hard:[word1,word2,...${words}]
    }
    NOTE 1:your answer should only return only the generated object in JSON format and no other text
    NOTE 2:respect the given language of the category and generate words in the same language
    `;
};

```

Figure 5-20. AI Word Generation Command Snippet

Along with the process the getWords function on the server side is responsible for sending a request to OpenAI GPT-3.5 using the constructed prompt. The response from GPT-3.5 is then processed to extract the generated words and returned to the client, as can been seen in Figure 5-21.

```
export const getWords = async (req: ExtendedRequest, res: Response, next: NextFunction) => {
  const { category, numberOfWords, difficultyLevels } = req.query;
  const prompt = Prompt(category, numberOfWords, difficultyLevels);
  const response = await openAi.chat.completions.create({
    model: "gpt-3.5-turbo",
    messages: [{ role: "user", content: prompt }],
  });

  const result = response.choices[0].message.content;
  return res.status(200).json({
    result,
    category,
    prompt,
    numberOfWords,
    difficultyLevels,
  });
};
```

Figure 5-21. Server Handling Generated Words Snippet

5.2.2.8.1 Benefits of OpenAI GPT-3.5 Integration

Dynamic Word Generation: OpenAI GPT-3 provides a dynamic solution for generating word pools based on user-specified criteria, enhancing the variety and richness of the gameplay.

Language and Format Control: The constructed prompt ensures that the generated words adhere to the language and format specified for the given category, maintaining consistency in the gaming experience

5.2.3 Hosting

5.2.3.1 Client Deployment With Vercel

For the client-side deployment of the Pictionary game, Vercel proves to be an excellent choice, providing seamless and efficient hosting solutions. The deployment process involves straightforward steps, and the Vercel platform offers a range of features to monitor and analyze the game's performance, as can been seen in Figure 5-22.

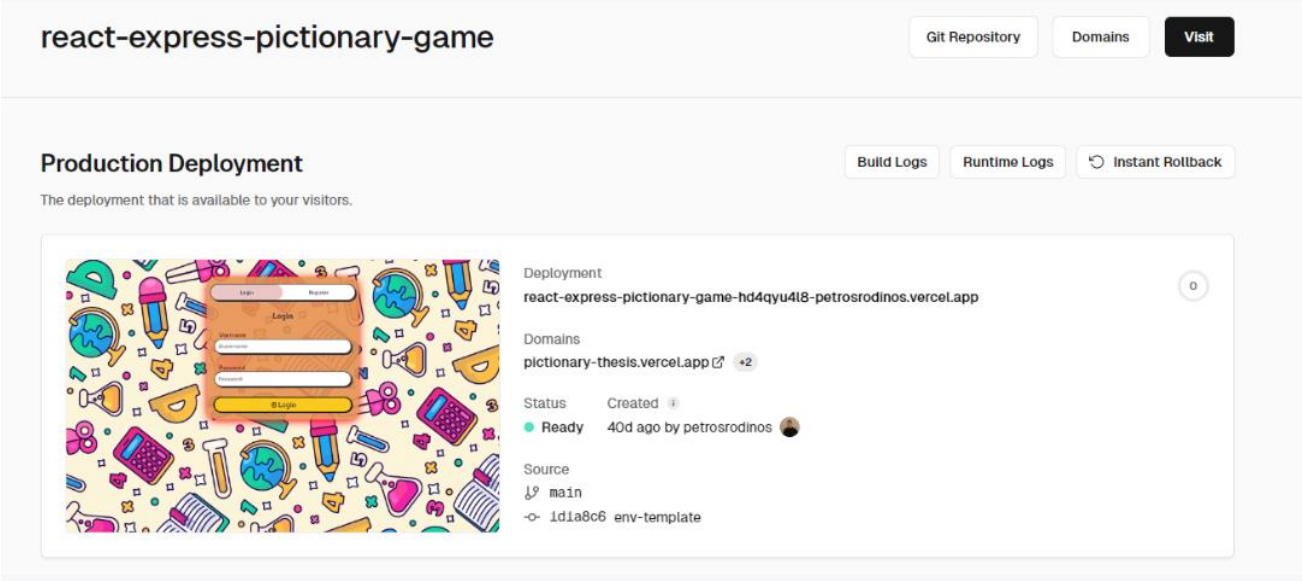


Figure 5-22. Versel Interface

As can been seen in Figure 5-23 Vercel's analytics page offers insights into the game's performance and user engagement:

Visitors: Tracks the number of unique visitors accessing the Pictionary game.

Page Views: Reflects the total number of page views, indicating user interactions and engagement.

Countries: Provides a geographical breakdown of user traffic, allowing for targeted analysis.

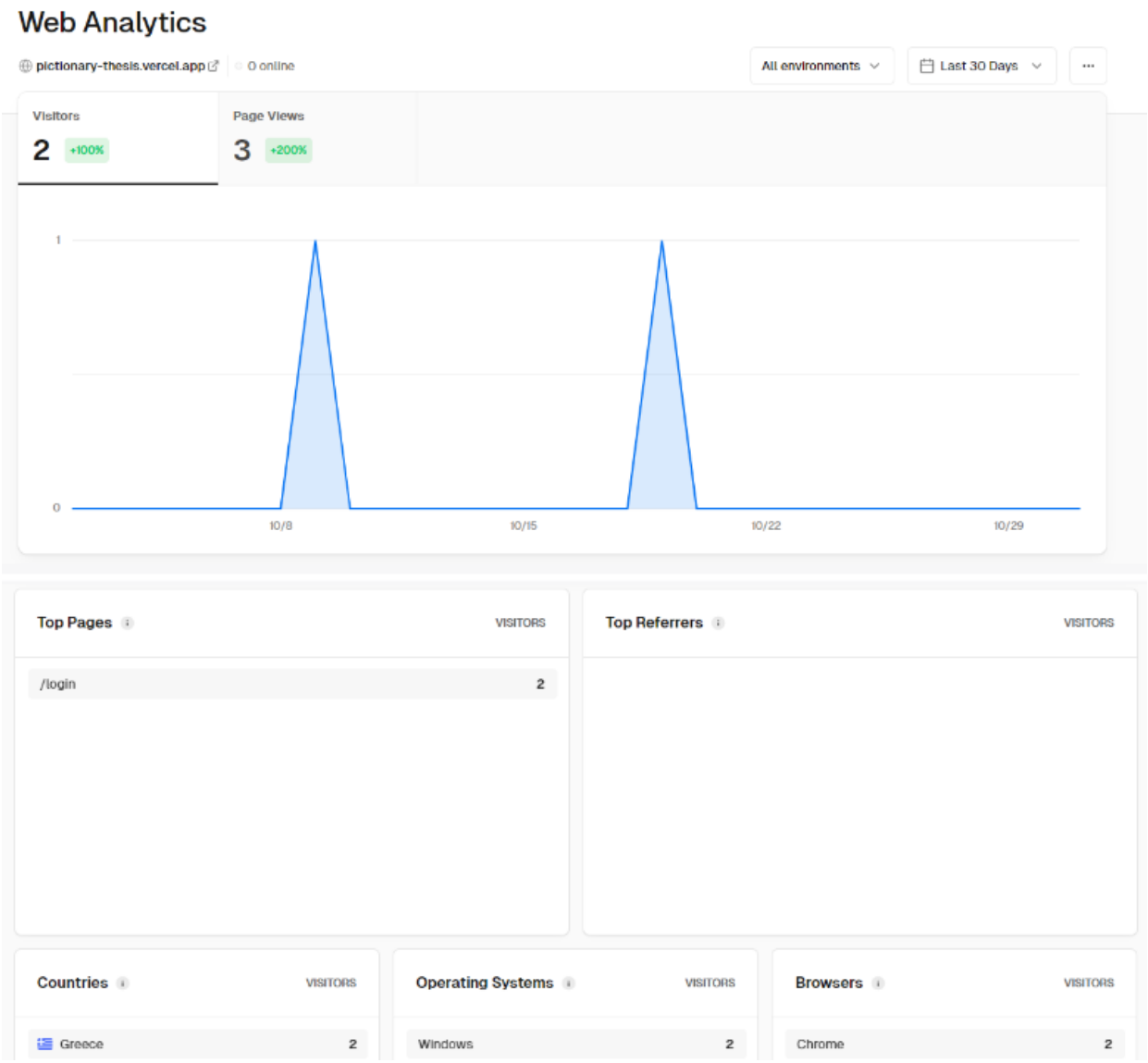


Figure 5-23. Versel Analytics

5.2.3.2 Server-Side Deployment with Heroku

Heroku serves as a reliable and user-friendly hosting solution for the server-side of the Pictionary game. The deployment page offers key information, such as deployment status and associated email, while also providing insight into local development commands called dynos , as can been seen in Figure 5-24.

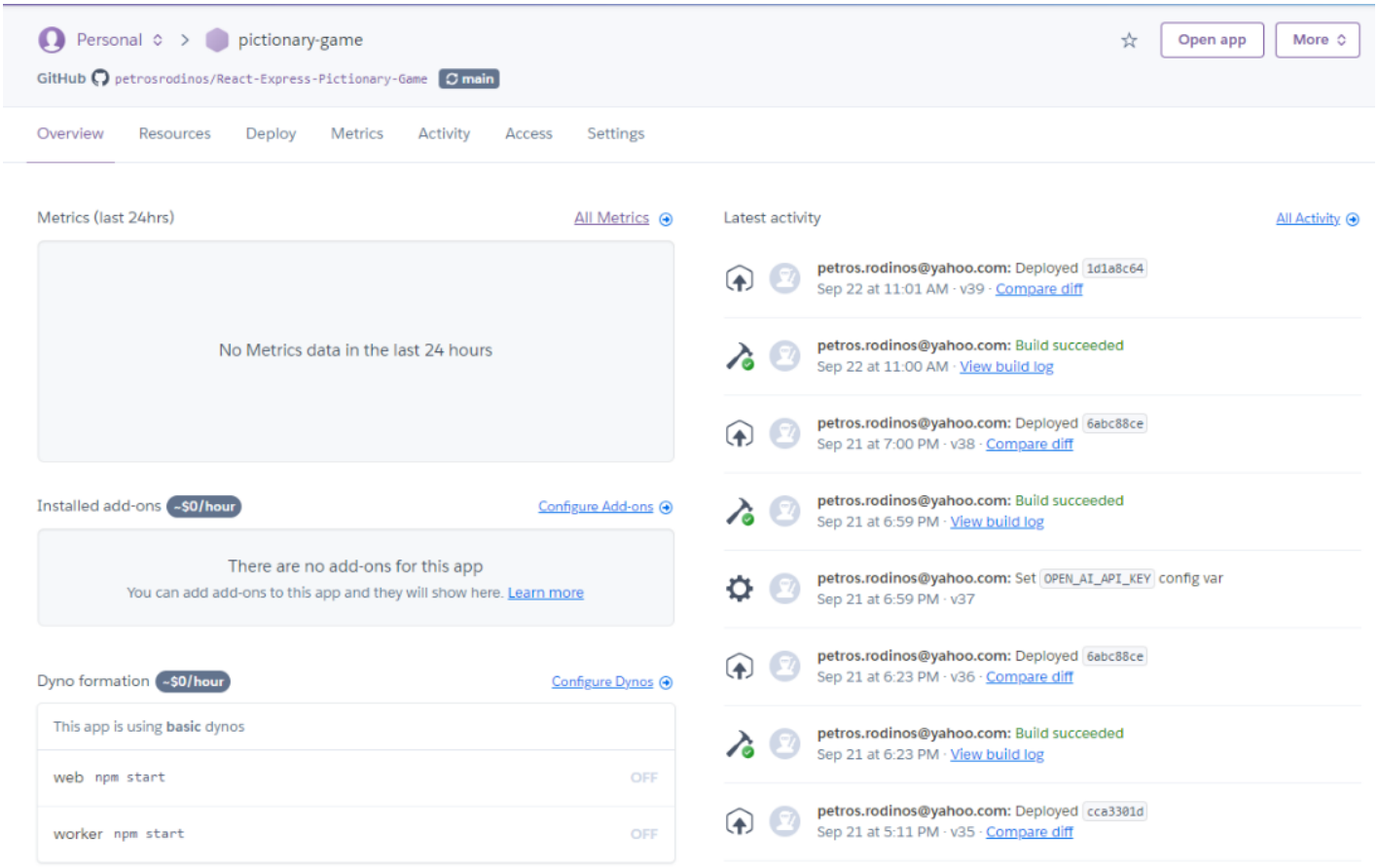


Figure 5-24. Heroku Interface

Chapter 6 - Result and Discussion

In this chapter, we embark on a journey through the fully realized application, showcasing the end product of our design and development efforts. Our presentation begins with the authentication page, providing an overview of the user experience as they navigate through the app.

6.1.1 Login/Register

The very first page that greets users upon opening the application is the login/register screen. If a user already possesses an account, they can seamlessly log in by entering their credentials navigating to the home page. If a user does not have an existing account, they can easily navigate to the register tab and initiate the account creation process. This user-friendly interface ensures a straightforward onboarding experience , as can been seen in Figure 6-1.

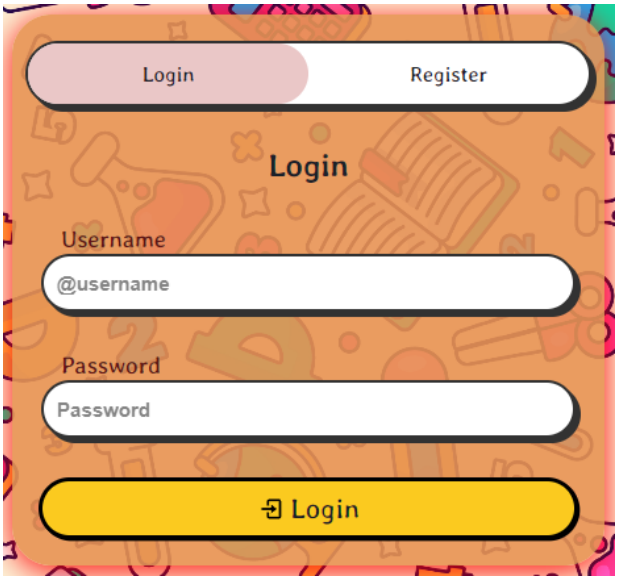


Figure 6-1. Login demo

6.1.1.1 Register

When navigating to the register page, users can create an account by providing essential information. This includes their chosen username, password, date of birth, role, and the selection of an avatar, as can been seen in Figure 6-2.

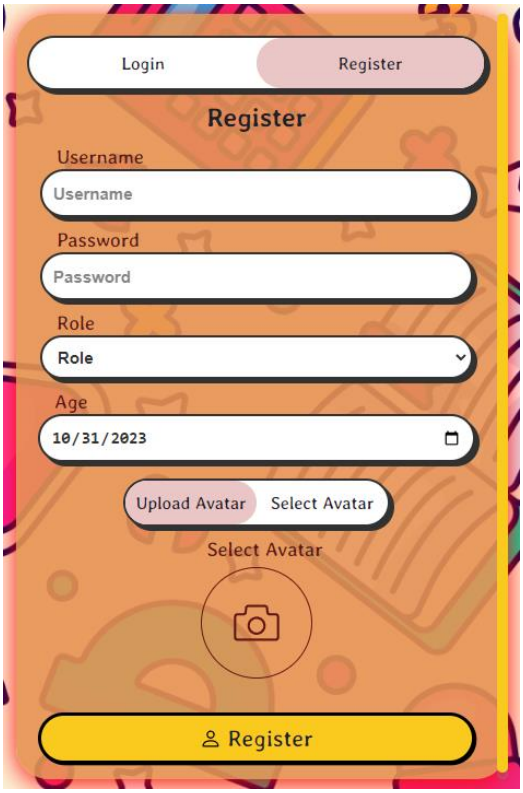


Figure 6-2. Register demo

Users can opt to select an avatar from the existing pool of images or upload their own preferred image to personalize their account, as can been seen in Figure 6-3 & Figure 6-4.

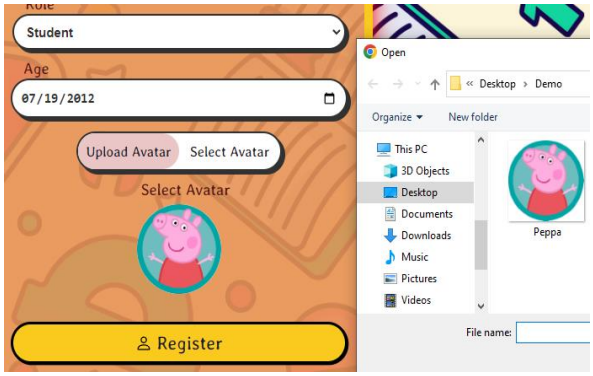


Figure 6-3. Upload Avatar demo



Figure 6-4. Select Avatar demo

6.1.2 Home page

As can been seen in Figure 6-5 upon successfully logging in or completing the user registration, users will be directed to the home page. In the home page, users will find a comprehensive set of features and information, ensuring a rich and immersive experience:

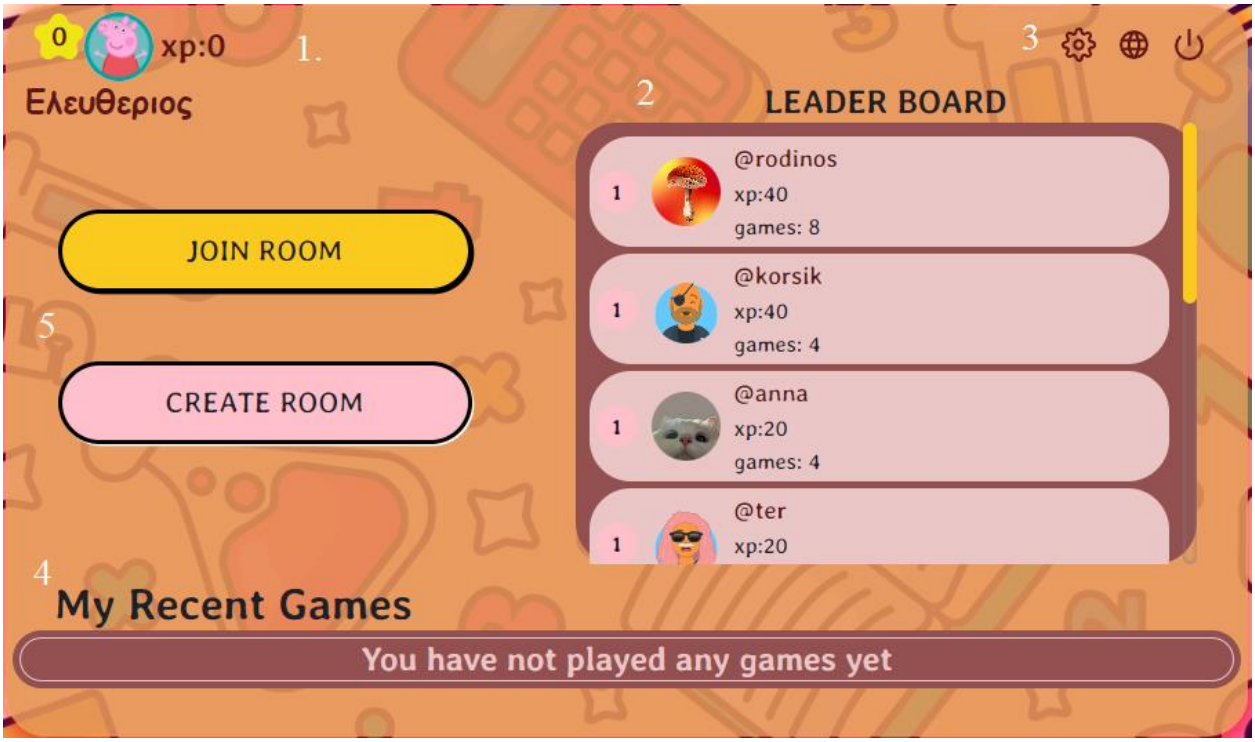


Figure 6-5. Home page demo

1. **User Profile:** At the top left, users will encounter their own avatar, along with their current level, icon, experience points (XP), and username. This personalized profile offers a quick overview of their progress within the game. In this demonstration account the level is 0 since games haven't been played yet.
2. **Global Leaderboard:** In the center-right section of the page, a global leaderboard is prominently displayed. This leaderboard showcases the rankings of all players who have engaged in the game, with the player holding the most points occupying the top position.
3. **Utility Buttons:** Positioned at the top right, users will find key utility buttons, including the settings button, language button, and the disconnect button, allowing for easy access to account customization and management.

Settings : If the settings button gets clicked a tab will pop up, on the top you can customize the game colors to your preferences .In the bottom of the tab you can change your profile credentials and avatar, as can been seen in Figure 6-6 & Figure 6-7.

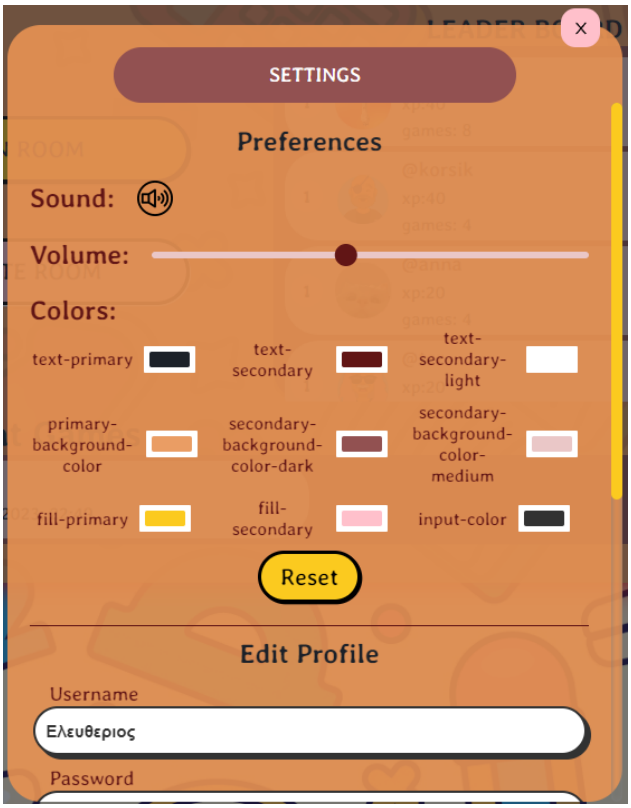


Figure 6-6. Settings tab top side

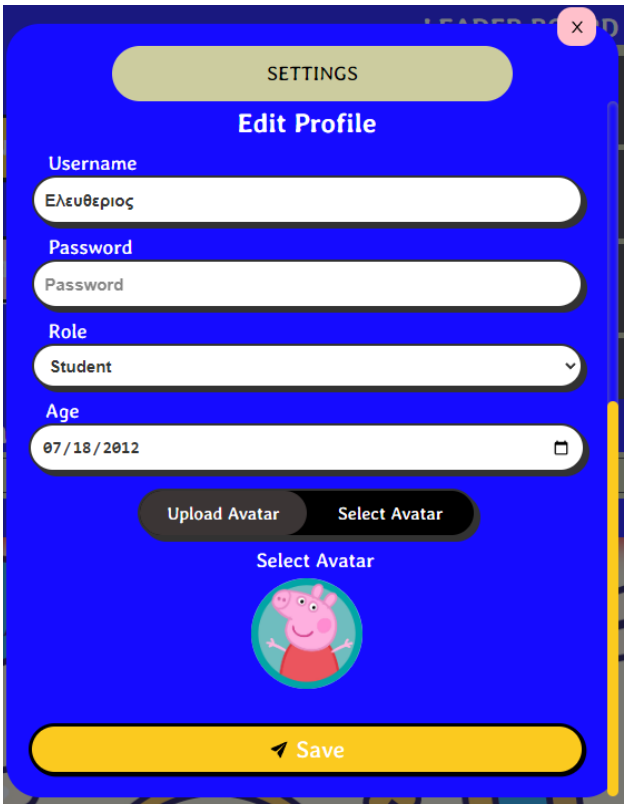


Figure 6-7 Customized settings tab bottom side

Language button : This button allows you to shift languages between Greek and English, as can been seen in Figure 6-8.



Figure 6-8 Language button

Disconnect button : Disconnects you from the game .

4. **My Recent Games:** Located at the bottom of the page, the "My Recent Games" section offers insights into the user's personal gaming history. It has information for the experience points gained from the game session and the exact date this game was played . It's important to note that for demonstration purposes, there may not be any recent games shown, as this account has just been created , but there will be soon after the demonstration of the game ends.

5. **Room Interaction:** To the center-left, users can readily access the "Join/Create Room" buttons, streamlining the process of initiating or joining game sessions with other players.

6.1.3 Room Creation

Upon clicking the "Create Room" button in the game, the user is presented with a comprehensive tab, offering a range of customization options for a personalized gaming experience. The tab features a unique room code, allowing players to either manually input it in the "Join a Room" section or effortlessly copy it via a provided link. Within this tab, players can fine-tune their game by selecting a specific word category to shape the theme of the drawing pool. Additionally, adjustable difficulty settings (easy, medium, hard) empower players to tailor the challenge level of the words they'll be drawing , as can been seen in Figure 6-9.

The customization options extend further with room parameters like size and the ability to set round time. Players can specify the number of participants required to commence the game, fostering flexibility in group dynamics. Moreover, players can configure the duration of each drawing round and the time allotted for word selection, adding an extra layer of strategic planning.

The amalgamation of these features holds significance as it enables players to tailor the game to their vocabulary level, allowing them to set personalized rules based on their preferences and skills. This customization ensures a dynamic and enjoyable gaming experience, accommodating individual abilities and gameplay preferences.

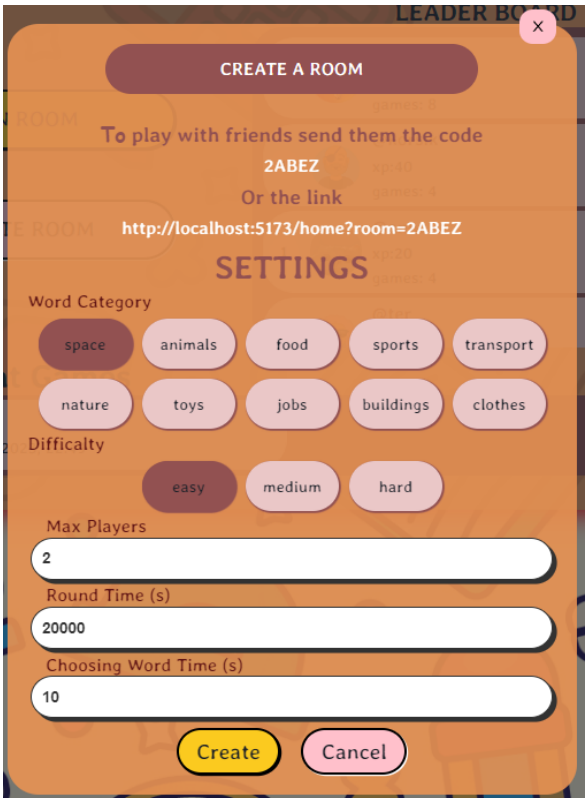


Figure 6-9. Room options

When a player eager to join a gaming session acquires a room code, they must proceed by clicking the "Join Room" button. This action allows them to input the received code and gain access to the designated room, as can been seen in Figure 6-10.

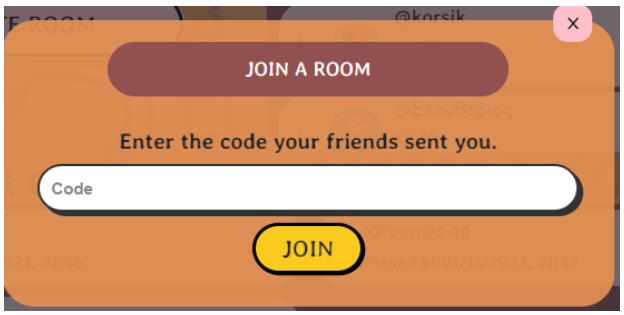


Figure 6-10. Join a room button

By doing so, a waiting tab will appear, displaying the list of other players who are also waiting to join the game. As the room space becomes fully occupied, a countdown timer will initiate, redirecting all players from the waiting area into the game once the timer reaches zero.

6.1.4 Gameplay

When the game starts, one player is randomly assigned the role of the sketcher, while the remaining participants take on the role of guessers. The sketcher is presented with a selection of five-word options within a specified category and must promptly choose a word to draw. It's worth noting that if the sketcher takes too long to choose, their turn will be skipped, ensuring that the game maintains pace, as can been seen in Figure 6-11.

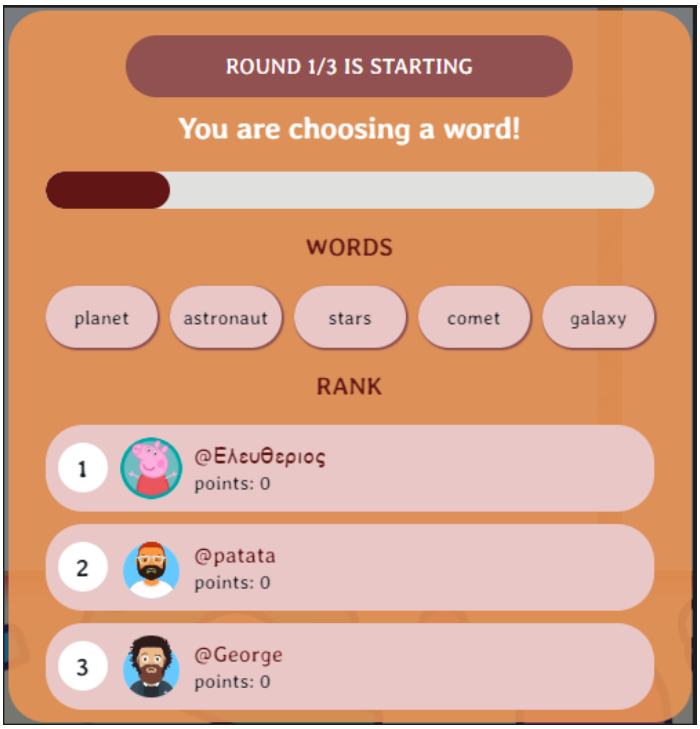


Figure 6-11. Choosing word

As can been seen in Figure 6-12 & Figure 6-13 while in the act of drawing, the sketcher has several creative tools at their disposal. These tools include the ability to select different colors, adjust the brush size, erase any mistakes, and even employ the camera to draw using their finger. This rich set of features empowers the sketcher to express their ideas with precision and creativity.

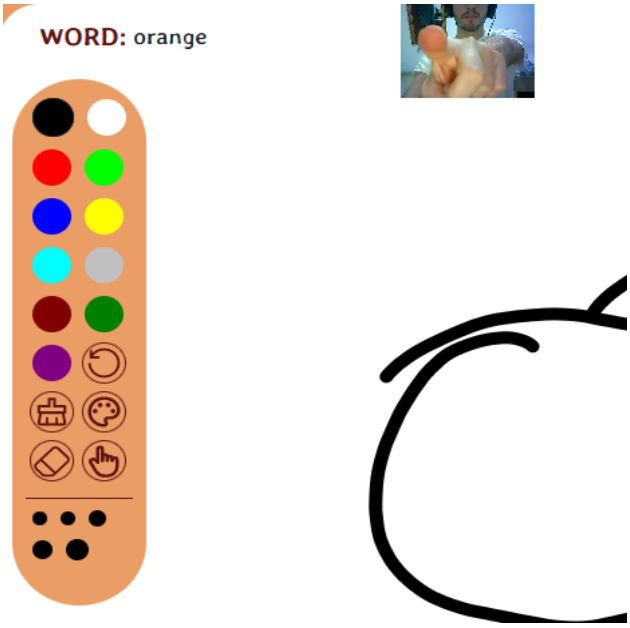


Figure 6-12. In Game Drawing tools

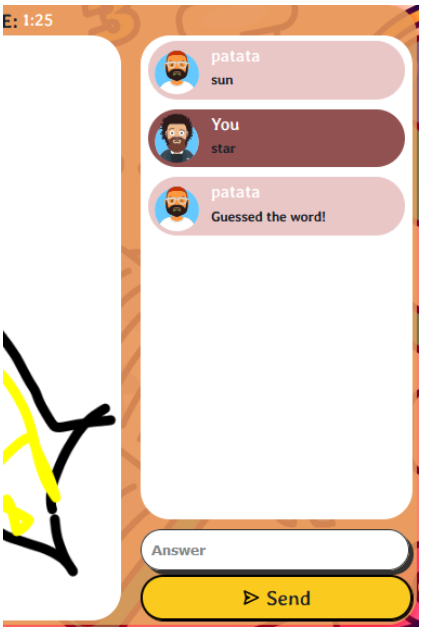


Figure 6-13 In Game Chat demo

While the sketcher is drawing, the guessers must actively engage by attempting to provide the correct answer for the depicted object. If a guesser successfully identifies the drawn item, their answer will be hidden, and they will no longer be able to submit further guesses for the remainder of the round. This mechanism ensures that each guesser has an opportunity to participate and keeps the game fair and balanced.

The process of changing between the sketcher and guesser roles continues until all allocated turns are exhausted, at which point the game concludes. all participants are redirected to the game finishing screen. Here, players can assess their performance relative to their opponents and gauge their progress in terms of leveling up within the game. This final screen provides a fitting conclusion to the gaming session , as can been seen in Figure 6-14.

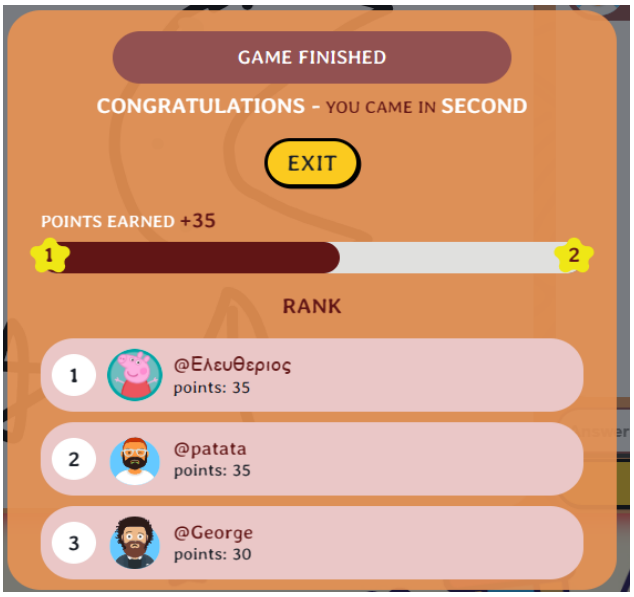


Figure 6-14. Game Finished Tab

Upon returning to the home screen, players can observe their avatar's progression, as well as any changes in their level and experience points (XP). This visual representation of progress allows players to track their advancement and showcases the dynamic nature of the game. Additionally, players can also review their updated match history, providing insights into their past gaming experiences, as can been seen in Figure 6-15.

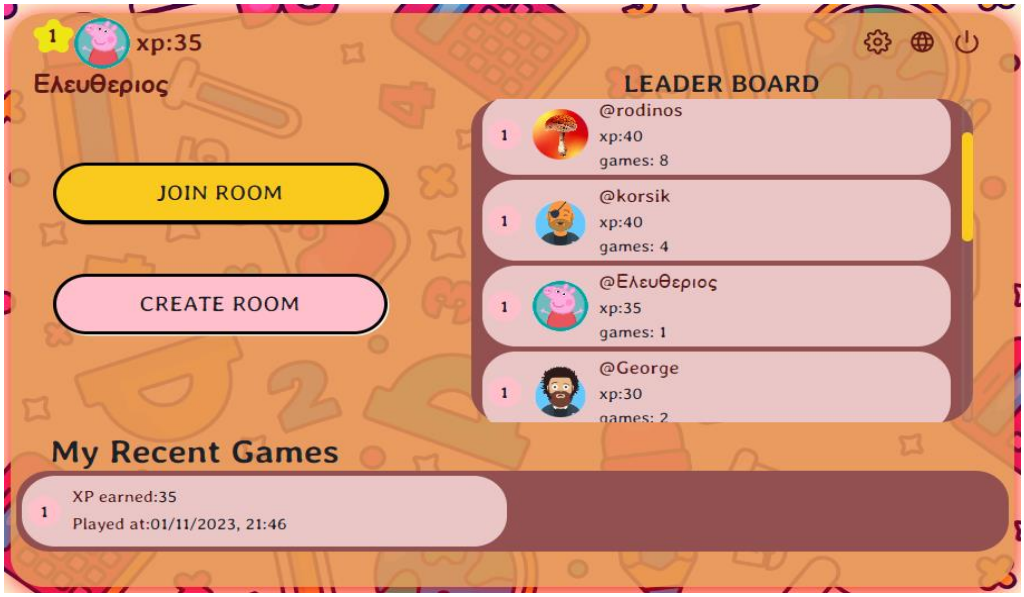


Figure 6-15. Updated home screen

6.2 Conclusion

In our journey of creating a Pictionary game, we succeeded in capturing the charm of the classic Pictionary game everyone knows and loves but also created a game intended for students who are trying to progress their vocabulary learning.

One notable stride is the creation of personalized user profiles, fostering a sense of ownership that resonates with the individuality of each player, creating an engaging environment that conduces learning. As users navigate through the homepage, the presence of a game leaderboard stands out, creating a beneficial dynamic for students. This competitive element not only injects a sense of excitement but also serves as a motivating factor, fostering a healthy spirit of competition among students. The leaderboard becomes an instrumental gauge for students to track their progress, amplifying their sense of accomplishment throughout vocabulary-learning. The customization options extend to the user interface, empowering students to craft an environment that aligns with their preferences. Such personalization doesn't merely cater to aesthetics but also contributes to an immersive learning experience. By providing flexibility in editing user credentials and images, we aim to create a space where students feel comfortable, fostering a positive association with the process of vocabulary acquisition. Educational value takes center stage when creating a game. The ability to select word pools and adjust difficulty levels allows students to tailor the vocabulary challenge to their proficiency, promoting targeted learning. This adaptive approach transforms the gaming experience into a practical language exercise, reinforcing vocabulary memorization.

The introduction of finger tracking for painting adds a layer of dynamism to gameplay, its potential impact on students is profound. The interactive nature of this feature could serve as a unique tool for educators, offering students a hands-on, engaging method to reinforce vocabulary learning and having fun at the same time. While the concept of finger tracking introduces an exciting prospect, it currently stands as a work in progress. Despite its functionality under optimal conditions, including good lighting, favorable camera angles, and decent camera quality, it falls short of entirely supplanting the conventional cursor for drawing. Acknowledging this limitation, our thesis recognizes the potential of finger tracking while candidly acknowledging the need for refinement.

The strengths of our Pictionary creation lie in the realm of personalization, user interface customization, engaging leaderboards, and the introduction of dynamic game mechanics. In

conclusion, our Pictionary game seeks to bridge the gap between entertainment and education, offering students a personalized, interactive, and dynamic platform to enhance their vocabulary skills. Through features crafted with the student in mind, we aspire to contribute to a positive and effective vocabulary-learning experience.

Chapter 7 - Future Work

In future iterations, several enhancements could elevate our Pictionary game, ensuring a richer and more adaptable user experience. First and foremost, enabling users to create and incorporate custom word pools would offer a tailored approach to learning, particularly beneficial for educators aiming to focus on specific word categories in their teaching. Expanding and updating the existing word pool to encompass a broader spectrum of vocabulary would further enrich the learning potential of the game.

Enhancing customization features to make the game even more engaging and entertaining for players could involve adding diverse themes, modes, and interactive elements.

Introducing additional language options beyond English and Greek would extend the game's reach, facilitating cross-cultural vocabulary exchange among students globally.

A notable future improvement could involve introducing a mode where players can compete against an AI opponent, leveraging the advancements in AI technology to create a challenging and dynamic gaming experience. Interestingly, the concept of AI in Pictionary games dates back to 1989 with “ Pictionary: The Game of Quick Draw” as mentioned in the chapter 2 of this thesis.

However, a critical aspect that demands attention in future developments is refining the finger tracking mechanics within the game. Overcoming limitations tied to factors like brightness and camera quality would significantly enhance the reliability and accessibility of this innovative feature, ensuring a smoother drawing experience for all users.

References Or Bibliography

[1] Y. Shabaneh, M. Farrah,“THE EFFECT OF GAMES ON VOCABULARY RETENTION”,2019

[2] A. Yustina Fatmawati, D. Miranty, W. Hammer, “THE EFFECTIVENESS OF PICTIONARY GAMES AS A TOOL TO TEACH VOCABULARY FOR SENIOR HIGH SCHOOL STUDENTS”,2023

[3] R. Angels , “Game Changer: The Story of Pictionary and How I Turned a Simple Idea Into the Bestselling Board Game in the World” , 2020

[4] B. Tinsman , “THE GAME INVENTOR’S GUIDEBOOK”,pp78, 2009

[5] B. Tinsman , “THE GAME INVENTOR’S GUIDEBOOK”,pp1-4, 2009

[6] Pictionary Bordgame , https://en.wikipedia.org/wiki/Pictionary#/media/File:Pictionary_Party.jpg , (accessed Oct 1, 2023)

[7] Pictionary , https://en.wikipedia.org/wiki/Pictionary#References , (accessed Oct. 24,2023)

[8] C. Scullion , “The NES Encyclopedia: Every Game Released for the Nintendo Entertainment System” ,pp 23, 2019

[9] Pictionary The Game of Video Quick Draw online emulator , https://archive.org/details/msdos_Pictionary_-_The_Game_of_Quick_Draw_1989 , (accessed Oct 29,2023)

[10] K. Guo, J. WoMa, E. Xu,“Quick, Draw! Doodle Recognition” , 2018

[11] Pictionary® Vs. AI , https://corporate.mattel.com/news/mattel-announces-pictionary-vs-ai-first-physical-board-game-to-integrate-ai , (accessed Oct. 27,2023)

[12] J. Guggemos, S. Seufert, “Teaching with and teaching about technology – Evidence for professional development of in-service teachers. Computers in Human Behavior”, https://doi.org/10.1016/j.chb.2020.106613 , 2021

[13] R. Scherer, T. Teo, “Unpacking teachers’ intentions to integrate technology:A meta-analysis. Educational Research Review”, https://doi.org/10.1016/j.edurev.2019.03.001 , 27, 90-109, 2019

[14] Y. Zhao, N. Wang, Y. Li, R. Zhou, “Do cultural differences affect users’ e-learning adoption? A meta-analysis” , British Journal of Educational Technology, https://doi.org/10.1111/bjet.13002 , 2020

[15] A. Granić, N. Marangoni, “Technology acceptance model in educational context: A systematic literature review”, British Journal of Educational Technology, 50(5), 2572-2593. https://doi.org/10.1111/bjet.12864 , 2019

[16] R. Scherer, S. K. Howard, J. Tondeur, F. Siddiq, “Profiling teachers' readiness for online teaching and learning in higher education: Who's ready? “ ,Computers in Human Behavior, 118, https://doi.org/10.1016/j.chb.2020.106675 ,2021

[17] S. Blömeke, T. Nilsen, R. Scherer,"School Innovativeness Is Associated With Enhanced Teacher Collaboration, Innovative Classroom Practices, and Job Satisfaction" ,Journal of Educational Psychology , https://doi.org/10.1037/edu0000668 , 2021

[18] I. K. R. Hatlevik ,O. E. Hatlevik, " Examining the Relationship Between Teachers' ICT Self-Efficacy for Educational Purposes, Collegial Collaboration, Lack of Facilitation and the Use of ICT in Teaching Practice" , Frontiers in Psychology, 9(935), https://doi.org/10.3389/fpsyg.2018.00935 , 2018

[19] F. Martin, T. Sun, C. D. Westine ," A systematic review of research on online teaching and learning from 2009 to 2018", Computers & Education, 159, https://doi.org/10.1016/j.compedu.2020.104009 , 2020

[20] K. Drossel, B. Eickelmann, J. Gerick,. "Predictors of teachers' use of ICT in school – the relevance of school characteristics, teachers' attitudes and teacher collaboration" , Education and Information Technologies, 22(2), 551-573, https://doi.org/10.1007/s10639-016-9476-y , 2017

[21] B.E. Penprase, "The fourth industrial revolution and higher education", Higher education in the era of the fourth industrial revolution 10 978–981, 2018

[22] V. Kryukov, A. Gorin," Digital technologies as education innovation at universities", Australian Educational Computing, 32, 1-16, 2017

[23] O. Lopez-Fernandez, "Emerging Health and Education Issues related to Internet Technologies and addictive problems", Int. J. Environ. Res. Public Health 18(1) , 321 , 2021

[24] A. Derakhshan, D.K. Elham," The Effects of Using Games on English Vocabulary Learning", Journal of Applied Linguistics and Language Research, 2015

[25] A. Helingo," Reading Habits and Integration of Technology in Understanding English Texts: A Case Study of EFL Students", Journal of English Language Teaching, Linguistics, and Literature Studies, 1(2),2021

[26] A. Rosidi, E. Faliyanti, "THE EFFECTIVENESS OF USING SCRABBLE AND BINGO GAME TECHNIQUE TOWARD STUDENTS' VOCABULARY MASTERY AT THE TENTH GRADE OF SMA N 2 METRO ACADEMIC YEAR 2013/2014", 40–49, 2017

[27] T. Prabha, A. A. Abdul "Effectiveness of Using Poly Category Mind Map for Vocabulary Development". Arab World English Journal, 11(2), 214–231, https://doi.org/10.24093/awej/vol11no2.15 , 2020

[28] K. Miyazaki, " The Effect of an Online Vocabulary Learning Tool on Passive and Active Vocabulary Use at a Range of Proficiency Levels". Journal of Pan-Pacific Association of Applied Linguistics, 23(1), 85–108, https://doi.org/10.25256/paal.23.2.5 ,2019

[29] T. M. Hoa, T. T. T. Trang "Effect of the Interactive Whiteboard on Vocabulary Achievement, Vocabulary Retention and Learning Attitudes", Anatolian Journal of Education,5(2), 173–186, 2020

[30] X. Ma, B. Yodkamlue," The effects of using a self-developed mobile app on vocabulary learning and retention among EFL learners", Pasaa, 58(December), 166–205 , 2019

[31] M. Selvi, A. Ö Çoşan," The effect of using educational games in teaching kingdoms of living things". Universal Journal of Educational Research, 6(9), 2019–2028. , https://doi.org/10.13189/ujer.2018.060921 ,2018

[32] Ian, Murray, " Use Of Multimedia Games For Biology Vocabulary Instruction", 2018

[33] M. R. T. Murga, G. F. L. González,L. A. N. Lira,"The games as a pegagogical tool in the improvement of its teaching practice", International Journal of Early Childhood Special Education, 12(1), 512–520. https://doi.org/10.9756/INT-JECSE/V12I1.201032 , 2020

[34] M. S. Bhatti, A. Habibie, S. Noreen, S. Hussain, S. Bajwa, " Realia-Mediated Instruction: An Effective Tool for Improving Young Learners' Vocabulary Skills", AlLisan, 6(2), https://doi.org/10.30603/al.v7i2.2176 , 2021

[35] R. D. Iswandari," The Effectiveness of Pictionary Game in Improving Students' Vocabulary Mastery a Quasi-Experimental Research of Fifth Grade Students of SD N 2 Ukir Rembang", 2017

[36] R. R. A. S. Bunau," The Effect Of Using Pictionary Games On Students",2018

[37] J. Sowell, "Talking about Words: A Vocabulary Description Game", In English Teaching Forum , 2018

[38] Z. Xu, G. Qiu, "DRAW-AND-GUESS: A MOBILE GAME WITH A PURPOSE FOR DRAWING BASED IMAGE RETRIEVAL", 2017

[39] J. Qian, T. Zhou, M. Young-Ng, J. Ma, A. Cheung, X. Li, I. Gonsher, J. Huang, "Portalware: Exploring Free-Hand AR Drawing with a Dual-Display Smartphone-Wearable Paradigm", 2021

[40] A. Kumar Bhunia, A. Das, U. Riaz Muhammad, Y. Yang, T. M. Hospedales, T. Xiang, Y. Gryaditskaya, Yi-Zhe Song, "Pixelor: A Competitive Sketching AI Agent. So, you think you can sketch?", 2020.

[41] R. Saxena, Z. Gaydos, M. Saaty, D. Haqq, P. Nair, G. Grutzik, Wei Lu Wang, J. Patel, "Fit to Draw: An Elevation of Location-Based Exergames", 2023

[42] https://dl.acm.org/doi/abs/10.1145/3573382.3616060 , (Accessed Nov. 7 2023)

[43] P. Xu, C. K. Joshi, X. Bresson, "Multigraph Transformer for Free-Hand Sketch Recognition", 2022

[44] J. Koushik," Understanding Convolutional Neural Networks", 2016

[45] R. M. Schmidt, "Recurrent Neural Networks (RNNs): A gentle Introduction and Overview",2019

[46] J. Villareale, A.V. Acosta-Ruiz, S. Adam Arcaro, T. Fox, E. Freed, R. C. Gray, M. Löwe, P. Nuchprayoon, A. Sladek, R. Weigelt, Y. Li, S. Risi, J. Zhu, "iNNk: A Multi-Player Game to Deceive a Neural Network",2020

[47] C. Clark, J. Salvador, D. Schwenk, D. Bonafilia, M. Yatskar, E. Kolve, Herrasti, J. Choi, S. Mehta, S. Skjonsberg, C. Schoenick, A. Sarnat, H. Hajishirzi, A. Kembhavi, O. Etzioni, A. Farhadi, "Iconary: A Pictionary-Based Game for Testing Multimodal Communication with Drawings and Text",2021

[48] N. Bansal, K. Gupta, K. Kannan, S. Pentapati, R. K. Sarvadevabhatla, "DrawMon: A Distributed System for Detection of Atypical Sketch Content in Concurrent Pictionary Games", 2022

[49] L. LEE ,C. Chau , K. Tsoi, N. Ling Yang , N. Wu , "A Mobile Game for Learning English Vocabulary with Augmented Reality Block Builder",2019

[50] M. Deshpande, B. Magerko, "Drawcto: A Multi-Agent Co-Creative AI For Collaborative Non-Representational Art" ,2021

[51] https://gvu.gatech.edu/research/projects/drawcto-multi-agent-co-creative-ai-collaborative-non-representational-art (accessed Oct 19 2023)

[52] Mozilla, https://developer.mozilla.org/en- US/docs/Learn/JavaScript /First_steps/What_is_JavaScript, (accessed Oct 15 2023)

[53] D. Crockford, JavaScript: The Good Parts, "O'Reilly Media, Inc.", 2008.

[54] Typescript: Documentation, "TypeScript for JavaScript Programmers?", https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html/ , (accessed Oct 15, 2023)

[55] Z. Parker , S. Poe , S. V. Vrbsky , "Comparing NoSQL MongoDB to an SQL DB",2013

[56] Ethan Brown, Web Development with Node and Express 2nd edition, "O'Reilly Media, Inc.," 2020

[57] E. Hahn, Express in Action: Writing, building, and testing Node. js applications. Simon and Schuster, 2016.

[58] Express, "Fast, unopinionated, minimalist web framework for Node.js." https://expressjs.com/ , (accessed Oct. 16, 2023).

[59] A. Mardan, Express. js Guide: The Comprehensive Book on Express. js. Azat Mardan, 2014.

[60] S. Aggarwal, "Modern web-development using React ," International Journal of Recent Research Aspects, vol. 5, no. 1, pp. 133–137, 2018.

[61] React official website, https://react.dev/learn/thinking-in-react/ ,(accessed Oct. 16, 2023).

[62] Developers - vercel/pkg Wiki [online] GitHub Wiki: https://github-wiki-see.page/m/vercel/pkg/wiki/Developers , (accessed Oct. 16 ,2023)

[63] Vercel Documentation, https://vercel.com/docs , (accessed Oct. 16, 2023).

[64] A. Hanjura, Heroku Cloud Application Development, 2014

[65] Heroku Documentation, https://devcenter.heroku.com/categories/reference , (accessed Oct. 16, 2023).

[66] Cloudinary Documentation, https://cloudinary.com/documentation , (accessed Oct. 16, 2023).
[67] OPENAI GPT text generation, https://platform.openai.com/docs/guides/text-generation , (accessed Oct. 16, 2023).
[68] SCSS Documentation , https://sass-lang.com/documentation/ , (accessed Oct. 16, 2023).
[69] Handtrack.js official website , https://victordibia.com/handtrack.js/#/ , (accessed Dec. 16, 2023).
[70] TensorFlow documentation , https://www.tensorflow.org/api_docs , (accessed Dec. 16, 2023).
[71] Dr N. Gibbins, “Cross-Origin Resource Sharing” , 2020