

RMIT University

School of Science and Technology (SST)

COSC2659 – IOS Development



Assignment 2 – Making Dream Game

Chess King – Crazy About Chess

Lecturer: Mr. Tom Huynh

Student: Ngo Ngoc Thinh – s3879364

Submission Due Date: 04 September 2024

Table of Contents

INTRODUCTION	2
OVERVIEW	2
MOTIVATION, INSPIRATION AND AUDIENCE	2
IMPLEMENTATION DETAILS	3
PROJECT STRUCTURE AND OVERVIEW	3
DATAFLOW	3
<i>ModelData: Centralized In-Memory Data Management</i>	4
<i>SessionManager: Managing User Sessions</i>	4
<i>FileStorageManager: Persistent Data Storage</i>	4
<i>Mock Data Initialization</i>	4
LOGIN AND REGISTER	5
MENU VIEW	5
PROFILE VIEW	6
HOW TO PLAY VIEW	7
SETTINGS VIEW	8
<i>Sound</i>	8
<i>Localization</i>	8
<i>Theme</i>	9
<i>AI Difficulty</i>	9
GAME PLAY VIEW	10
<i>Scoring System</i>	10
<i>Badges System</i>	11
<i>Game Play and State</i>	11
<i>Game Continuity</i>	12
LEADERBOARD VIEW	13
ANIMATION	14
KNOWN BUGS	14
DESIGN ELEMENTS AND USER EXPERIENCE	14
<i>Visual Appeal</i>	15
<i>Intuitive User Interface</i>	15
<i>Consistency</i>	15
<i>User-Centered Design</i>	15
CONCLUSION	16
REFLECTION	16
FUTURE IMPROVEMENTS	16
REFERENCE	17
APPENDICES	18

Introduction

The Chess King mobile application is a comprehensive platform that fully engages users in the strategic world of chess by means of an amazing UI, advanced gameplay capabilities with multiple playing modes, and a robust emphasis on community engagement. My application is more than a mere game; it is a digital environment designed for chess enthusiasts who strive to enhance their abilities and push their limits.

Overview

Chess King offers a wide variety of features specifically tailored to accommodate players of all skill levels. The game features a scoring system, achievement badges, highly customizable themes, captivating sound effects, and challenging AI opponents. The app's icon, featuring a Rook, symbolizes strength, strategic depth, and calm, reflecting the qualities of a skilled chess player. This selection highlights the app's focus on enabling users to engage in critical and strategic thinking.



Figure 1. Application's icon.

Motivation, Inspiration and Audience

Motivation: My enthusiasm for chess originated during my childhood when I played on traditional boards, which then shifted to digital platforms with the advancement of technology. The mentioned transition underscored the potential of technology to enrich this age-old game by introducing more features such as interactive elements and adaptive challenges that the traditional game can not bring to the table. Chess King was born with the aim of combining classic chess with modern digital cutting-edge technologies and strives to provide a customized chess-playing experience and encourage user competitiveness.

Inpiration: Chess is also an intellectual activity. It develops critical thinking and strategic planning for life beyond the game. Recognizing these benefits and inspired by my own chess adventure, I created an app that acts as a place for playing but also for learning through tutorials embedded in the app.

The game engine gets inspiration from a seasoned IOS developer [1].

Audience: Chess King is for everyone, ranging from my friends and family, with whom I initially shared and tested the game, to the online chess community after the course. From casual players who enjoy occasional games to tough challengers who love competition can find these places in the app. Rather than just another chess game, Chess King is a user-centric gaming experience that respects chess's traditional rules and embraces mobile entertainment.

Implementation Details

Project Structure and Overview

The project files are systematically organized into several categories [Appendix 1] to enhance efficiency during application development and for future expansion and maintenance. Specifically, there are multiple protocols defined within the application so that the maintainer can easily extend the application, an example of this is the ArtificialIntelligence protocol requires to implementation of a function called “nextMove(game: Game)” that we pass in the current game and for any new AI player the mission of developers are to implement business logic for this function to return a next move. This use of protocol as a generic constraint follows Apple’s best practice [2].

Dataflow

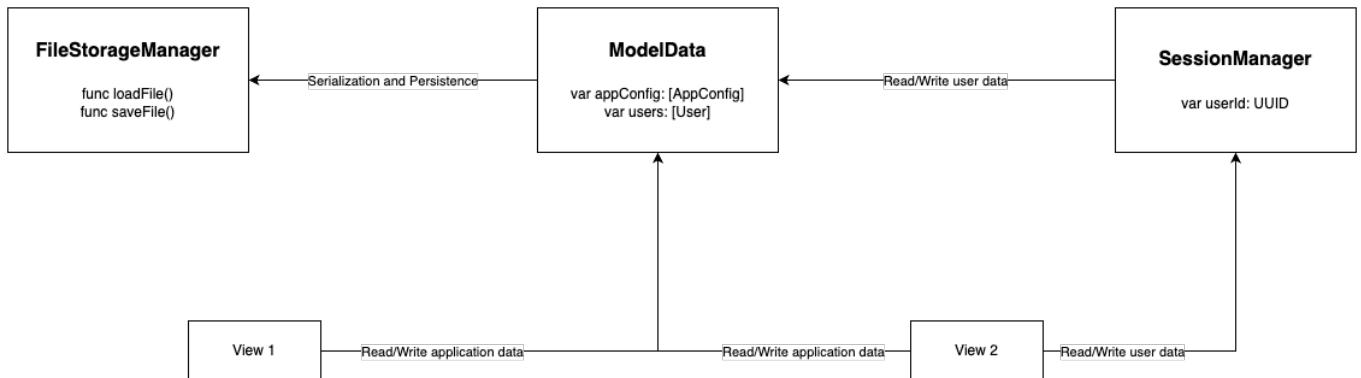


Figure 2. Dataflow.

ModelData: Centralized In-Memory Data Management

The ModelData functions as the centralized in-memory data storage for Chess King, serving as the single source of truth (SSOT) for the whole application. By adhering to the observation framework [3], this ModelData guarantees that any modifications made to the data are promptly displayed in the user interface, eliminating the need for a restart.

SessionManager: Managing User Sessions

The main objective of the SessionManager component is to manage and regulate user data across the application such as login, register, save the user's game history,.. Actually, SessionManager manipulates user data inside ModelData (remember ModelData is SSOT), but this segregation is a need and guarantees that ModelData exclusively concentrates on application-specific configurations such as theme and music, while SessionManager handles user-specific information such as gaming history and current session parameters.

FileStorageManager: Persistent Data Storage

In addition to the in-memory storage offered by ModelData, FileStorageManager provides persistent data storage functionalities. This class guarantees that any modification made to ModelData is converted into JSON format and stored on the device's storage. This enables the program to be resilient throughout restarts. The highly flexible design of the communication between ModelData and FileStorageManager allows for a possible replacement to accommodate various storage locations, in the scope of this assignment only on the iOS file system as supplied by FileManager [4].

Mock Data Initialization

In scenarios where no prior data is available, such as on the first launch, the application generates mock data to simulate a realistic initial state. The process for generating this data is encapsulated in the “User.generateMockUsers()”, which strives to create authentic initial mock users, but some parameters still do not align with real-world scenarios due to their randomized nature such as scoring history.

Login and Register

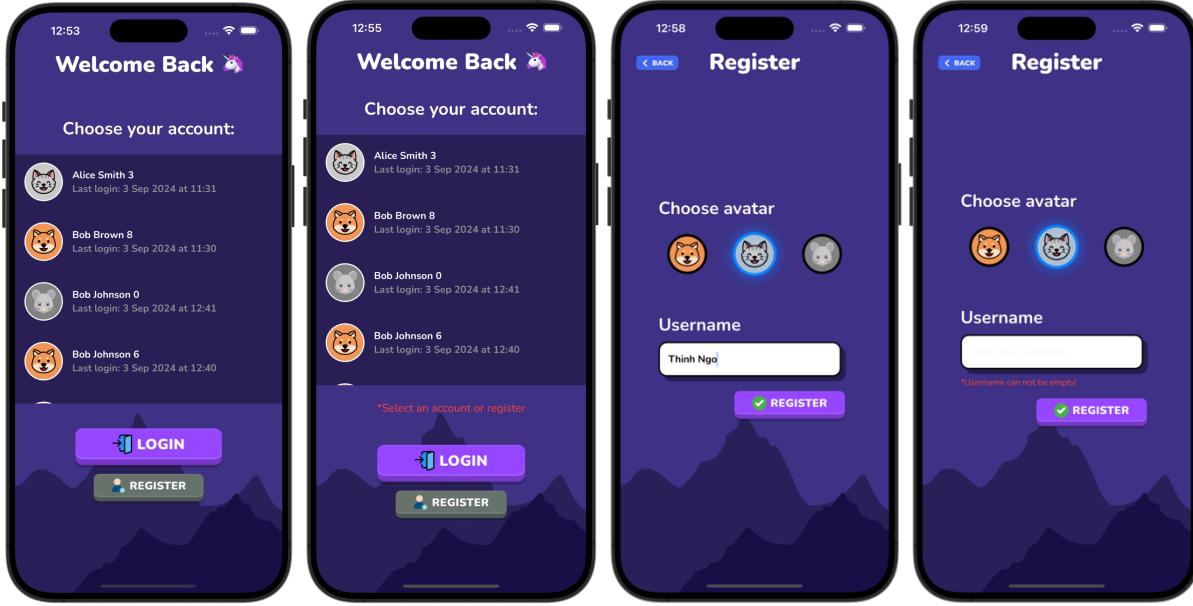


Figure 3. Login and Register view.

The main focus of the application is directed at gameplay, so authentication techniques, particularly password protection, are excluded in order to simplify user access. Users are offered the option to choose an existing account from a supplied list on the welcome screen or create a new account without the need to provide a password. Both the login and registration procedures provide thorough error validations with suitable feedback messages, such as "username already exists" and "username cannot be empty". Following a successful selection or creation of an account, users are smoothly redirected to the Menu view, enabling instant interaction with the game. The system will allow the user to one of three avatars dog, cat, or mouse. For better visualization please refer to the [demo video at 0:12](#).

Menu View

The Menu screen functions as the central interface [Figure 4], enabling user navigation to features including playing games versus AI, 2-player modes, and accessing user profiles via the top-right of the screen,... The figure shown above depicts the Menu View in two separate application themes: Halloween (left) and Modern (right). For a thorough examination of the user interfaces within both themes, refer to the theme section of this document. Henceforth, this document will exclusively showcase the UI in the Halloween theme, which is also the default theme upon initial installation of the application. For better visualization please refer to the [demo video at 0:41](#).



Figure 4. Menu view in Halloween and Modern theme.

Profile View



Figure 5. Profile views: Stats, Badges and Score History.

In order to provide users with a dedicated personal space to track their statistical data and growth over time, the Profile View has been introduced. The interface enables users to modify their avatar by using the pencil symbol positioned at the lower right corner of the avatar display, therefore initiating an avatar selection dialogue, which is implemented with a

Sheet view. Subsequently, on the Profile View, users are shown a list of their acquired badges (see the Game Mechanism section for specific information on badge categories) with spring animation ([demo video at 1:54](#)). An interactive line chart presented at the bottom of the Profile View illustrates the variations in the user's score over time and user's average scores ([demo video at 2:02](#)). This graph serves the purpose of not only visualizing the individual development but also enabling comparison with the standings of other players.

How To Play View

An essential feature in the Chess King app, the "How to Play" view [Figure 6] serves to acquaint beginners with the basic principles of chess and to improve the abilities of seasoned players with tips and tricks. A total of 14 steps from [5] represented 14 pages that direct users from fundamental principles to more complex strategies and movements such as castling, en passant, and checkmate scenarios. Users have the ability to proceed between steps or revert back to the menu view by utilizing three buttons positioned at the lower section of the interface. Transition animations visually enhance the movement from one step to the next, enriching the learning experience, and ensuring concepts are not only read but seen (refer to [demo video at 1:14](#)). In addition, all the content is supported with localization of English and Vietnamese.

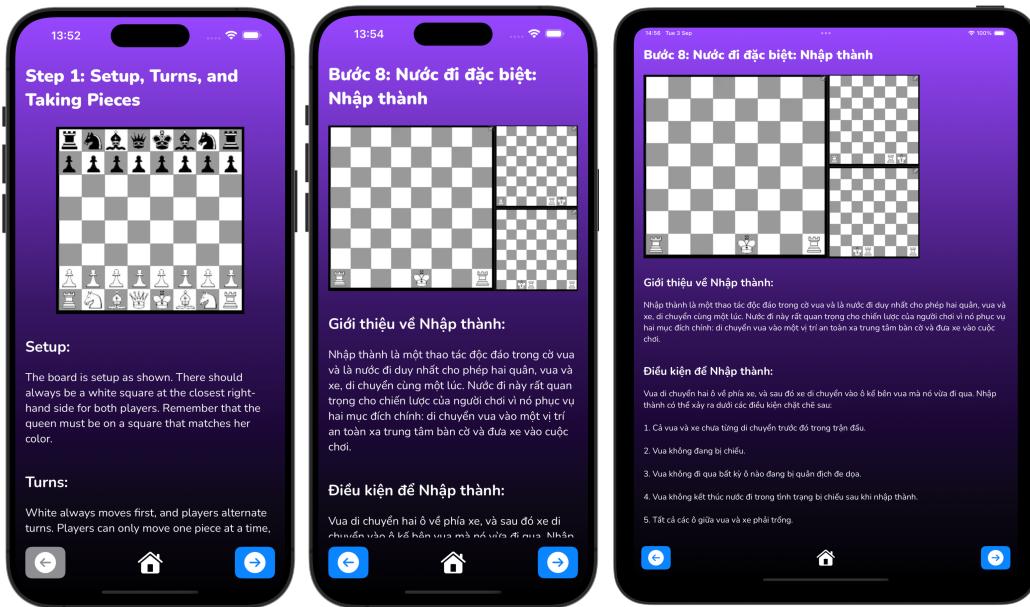


Figure 6. How to play view on iPhone and iPad.

Settings View



Figure 7. Settings view: Sound, language, theme and AI difficulty.

Sound

The application incorporates a diverse of background music and sound effects across various views and game states to enhance the user experience. The sound can be turned on/off in the settings view by tapping in the sound option and the settings will take effect immediately without the app restart [Figure 7].

- Each primary view: Menu, Leaderboard, and Game has its own background music, setting the appropriate mood for each section.
- Interactive sound effects accompany any button clicks and in-game actions to add tactile feedback to user interactions.
- Special audio occurs to signal game events like wins or losses, heightening the emotional impact of the game.

Localization

The application enables localization by providing support for both English and Vietnamese languages in all views. Users may choose their preferred language by selecting the "Language" option in the settings. Subsequently, a pop-up will be displayed, enabling them to select their desired language [Figure 7]. The previously mentioned ModelData architecture

enables the language settings modifications to take effect immediately (refer to [demo video at 4:10](#)).

Theme

To appeal to a wide variety of visual tastes, the application provides two separate app themes: Halloween and Modern; along with three distinct chess board themes: Halloween, Modern, and Wood. A total of six possible theme combinations enables players to personalize their experience by selecting from six possible theme combinations, therefore satisfying even the most picky taste [Figure 7]. For a detailed exploration of the various theme configurations available within the app, please refer to [demo video at 3:36](#).

AI Difficulty

Incorporates different AI difficulty settings: easy, medium, and hard [Figure 7]. Designed to cater to a wide range of skill levels, ensuring a challenging and engaging experience for all types of players.



Figure 8. From left to right: Michael, Alex and TomHuynh.

- Easy-level “**Michael the dumb AI**”: Well-suited for beginners and can play offline. His brain is actually made of the Minimax algorithm with alpha-beta pruning and the algorithm is implemented with Swift following this tutorial [1], [6]. The move is actually calculated using the device’s CPU so it is a little bit slow and again this easy level is not actually easy for beginners, readers should try the easy one first.
- Medium-level “**Alex the clever AI**”: Designated for players with intermediate skill levels, his brain is made of the well-known Stockfish API [7] with a search depth of 10 to simulate an estimated Elo rating of around 1800 [8].
- Experienced chess players looking for a difficult challenge might find boring with Michael and Alex, then, allow me to introduce to you “**TomHuynh the super AI**” crafted using Stockfish API with a search depth of 15, equating to an impressive Elo

approximately 2563 [8]. Interestingly, even I, the developer, find myself regularly overwhelmed by this AI level. Therefore, if you think you are good at chess, I encourage readers to test your skills against TomHuynh and share your results.

Note: Reader! If you want to take on Alex or TomHuynh, make sure you are connected to the internet - these guys cannot get out of bed without it! If you are offline, they will just groggily push pieces around the board at random.

Game Play View



Figure 9. Game play in iPhone and iPad.

The application is made with love to adjust every little detail that makes it look decent on both iPhone and iPad. Users can play the game in two modes by clicking the appropriate button in the menu view: Play versus AI or play with a different user by entering another username.

Scoring System

In Chess King, players earn points not just for their winning but get punished when they lose to enforce the competitive ranking within the game:

- **Regular Wins:** Bag 100 points for any win. And minus 100 points for any lose, the score cannot be negative so it will be reduced to zero if not enough points to reduce.

- **AI Conquests:** Tackle AIs for extra scores. Defeat Easy AI for an extra 300 points, Medium AI for 500 points, and Hard AI for a whopping 700 points on top of winning. Note that extra scores only apply in versus AI mode.

The players' scores will be updated for every completed game and be updated in the line chart in the player's profile. AIs also be treated as real players so their scores after a game will also be updated accordingly.

Badges System

In Chess King, every major achievement earns you a badge, turning every game into a quest for glory. There are a total of 8 badges players can earn, for the list of all badges and conditions please refer to [Appendix 2]:

Game Play and State

During gameplay, the chessboard is centrally displayed, with each player's information: name, winning rate, and avatar positioned above and below the board respectively [Figure 9]. By choosing a chess piece, players begin a move, upon which the game presents all possible legal moves for that piece [Figure 9]. Most importantly, there is a limitation when the king is checked, the game limits possible moves to only those that can reduce this danger, therefore assisting beginners in avoiding typical mistakes. In the event of a checkmate, the game discontinues without delay, and a comprehensive statistics pop-up is displayed, displaying changes in scores, counts of special moves executed, and any newly acquired badges [Figure 9]. This special limitation improves the process of learning by avoiding premature terminations of the game.

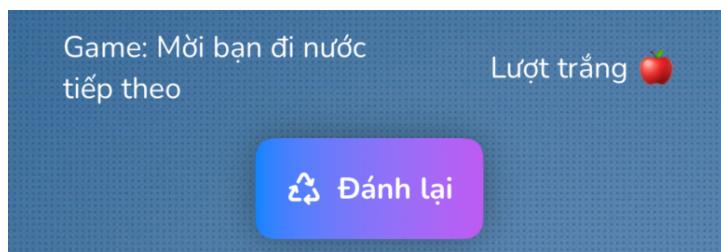


Figure 10. Game status display.

The game status is prominently displayed beneath the board, providing players with crucial information about the current state of play. For players uncertain about where they are in the game, the left side of the game status bar offers guidance on potential actions. Conversely, the right side of the status bar displays the current player's turn, either black or white. In

In addition, A prominent button at the bottom of the screen lets players reverse moves. This feature reverses the player's last move and the AI's before it, making it beneficial in AI matches to let players correct mistakes to continue improving. Especially, there is a game state when the AI is making a move, the game status will display such thing “**TomHuynh the super AI: I'm thinking**” which makes us feel like we are playing against a human. For more visualization, please refer to the [demo video at 4:48](#).



Figure 11. From left to right: Promotion, En Passant and Castling move.

Special moves such as promotion, en passant and castling are supported out of the box. Note that if the user chooses to cancel the promotion the queen will be chosen by default.

Game Continuity

In the event that a user attempts to leave [Figure 12] the game by using the back button, a confirmation notification helps to prevent accidental termination. Returned to the menu view, a red button notifies users of an active game that they can resume. By serializing the game state as JSON, this feature remains functional even when the app is fully closed and reopened again enabling users to resume progress exactly where they left off.



Figure 12. Game state retained with resume button.

Leaderboard View



Figure 13. Leaderboard and badge statistics.

The leaderboard is divided into two sections for enhanced player competitiveness:

- **Top player ranking:** This section displays the names, scores, and main badges of the top 10 players ranked by score. In order to see a player's profile, users can just tap on their row, which is identical to the personal profile view but without the ability to customize avatars.
- **Badge Distribution Analytics:** Displays an interactive bar chart that emphasizes the limited number of rare badges among the highest-ranking players. This graphical depiction highlights the limited availability of specific rare badges, therefore inspiring players to engage in additional gameplay to achieve them. Note that rare badges are just a concept, the mock data does not actually reflect the rarity.

Animation



Figure 14. Button states.

Each button integrates spring animations, which offer visual feedback upon pressing or holding. Additionally, navigation throughout the app utilizes the NavigationStack, enabling slide transition animations between views.

Known Bugs

Despite considerable effort to ensure robust game state serialization, issues persist that occasionally lead to crashes upon resuming a game. While the current implementation allows game states to be restored, inconsistencies in the serialization process occasionally result in unstable game behavior. Therefore, addressing this problem will be a primary focus for future enhancements.

Design Elements and User Experience

There are several aspects that I have considered when making the application that must have a high level of user interaction and support both multiple theme combinations:

Visual Appeal

The design of Chess King is thoroughly hand-picked and crafted from the smallest detail like buttons, badges, logos,.. in various game designs from Figma [9], [10] and [11] ensuring an appealing visual experience across both light and dark modes. There are two accent colors (#9747FF and #4068F5) are used to highlight interactive elements and important information aiding navigation and usability. Nunito font [12] is the main font used across the entire application. The application also uses some built-in components from SwiftUI such as NavigationStack or Sheet to bring an authentic IOS vibe to the user but also add my own design to make the application unique such as theme customization.

Intuitive User Interface

Chess King features a user-friendly UI specifically created to enable effortless navigability by using and placing navigation elements in the familiar position to the user such as the back button, and popup view,.. When there is a need for attention, there will be a highlighted UI component for the user to notice such as the game resume button. In addition, by incorporating size-responsive elements, the interface guarantees efficient navigation in both small and large size screen devices.

Consistency

Maintaining consistency is crucial in the design of Chess King, uniform design components, including typefaces, button designs, and color schemes, are carefully integrated into all views. By utilization of built-in IOS icons, UI components, and creation of customizable reusable UI components to enforce the consistency between different sections within the application. Moreover, reusable UI components make the development of theme customization blazingly fast and enjoyable. In the future, if the contributors wish to add more themes into the application all they need is to add a new entry in an enum and import the needed image assets.

User-Centered Design

Chess King is designed with the user as the central focus, prioritizing flexibility and customization. Gamers have the ability to personalize their gaming experience by selecting customized themes and adjusting difficulty settings, all conveniently controlled within the app's user-friendly interface. From day one, I have considered for this feature, that the

system's responsiveness must guarantee the instantaneous effect of setting changes such as theme modifications or game configuration adjustments, therefore increasing user satisfaction. Finally, Chess King is specifically created to accommodate both casual enthusiasts and experienced players.

Conclusion

Reflection

The development of Chess King was highly educative and a wonderful journey, this project enhanced my proficiency in UI design using Swift and SwiftUI, emphasized the need for thorough testing, consideration, and examination of user favorites, and also refined my comprehension of dataflow following Swift way.

Background music and sound effects: I learned that music and sound effects really boost the user experience 100% with just a very small effort compared to developing the whole game.

API integrations: There is a dose of public and free APIs for our next application, an example of this is our Chess King can entirely use public APIs to create an AI-integrated game without the need for in-depth knowledge about AI.

Future Improvements

Advanced AI: Implementing an adaptive AI that can adjust the difficulty to different user skill levels that could make the game more challenging and engaging.

Online Matching: The incorporation of an online matching feature would enable worldwide real-time competitions therefore greatly enhancing the chess community involvement.

Bug Fixes and Performance: Continual efforts to resolve serialization conflicts and enhance the overall stability of the application.

Reference

- [1] YouTube, <https://www.youtube.com/watch?v=Va1Xeq04YxU&t=15559s> (accessed Sep. 3, 2024).
- [2] “This page requires JavaScript.,” Swift.org, <https://docs.swift.org/swift-book/documentation/the-swift-programming-language/protocols/> (accessed Sep. 3, 2024).
- [3] “Observation,” Apple Developer Documentation, <https://developer.apple.com/documentation/observation> (accessed Sep. 3, 2024).
- [4] “FileManager,” Apple Developer Documentation, <https://developer.apple.com/documentation/foundation/filemanager> (accessed Sep. 3, 2024).
- [5] Chess. com Team, “How to play chess: Learn the rules & 7 steps to get you started,” Chess.com, <https://www.chess.com/learn-how-to-play-chess> (accessed Sep. 3, 2024).
- [6] “Implementing minimax with alpha-beta pruning for a simple chess AI in swift,” Implementing Minimax with Alpha-Beta pruning for a simple Chess AI in Swift, <https://web.navan.dev/posts/2024-04-17-implementing-minimax-for-chess-in-swift.html> (accessed Sep. 3, 2024).
- [7] “Stockfishonline: Stockfish rest api,” StockfishOnline | Stockfish REST API, <https://stockfish.online/docs.php> (accessed Sep. 3, 2024).
- [8] “Stockfish Elo vs search depth,” Chess Stack Exchange, <https://chess.stackexchange.com/questions/8123/stockfish-elo-vs-search-depth> (accessed Sep. 3, 2024).
- [9] Rank badge design - free | figma, <https://www.figma.com/community/file/1358718222712048791/rank-badge-design-free> (accessed Sep. 3, 2024).
- [10] Interactive game UI | 3D button | figma, <https://www.figma.com/community/file/1362755493555595250/interactive-game-ui-3d-button> (accessed Sep. 3, 2024).
- [11] Input field forms | figma, <https://www.figma.com/community/file/1127625426899459390/input-field-forms> (accessed Sep. 3, 2024).

[12] “Nunito,” Google Fonts, <https://fonts.google.com/specimen/Nunito> (accessed Sep. 3, 2024).

Appendices

```
. └── Teris_MafiaApp.swift          -- Application
    └── Views                      -- Main views
        ├── ContentView.swift      -- Root view
        ├── MenuView.swift
        ├── GameView.swift
        ├── HowToPlayView.swift
        ├── SettingsView.swift
        ├── ProfileView.swift
        ...
    └── Components                  -- Reusable components
        ├── CircleButton.swift
        ├── GradientBackground.swift
        ...
    └── Models                      -- Model related to App's operation
        ├── App
            ├── User.swift
            ├── AppConfig.swift
            ...
        └── ChessBoard                -- Model related to Game's operation
            ├── Game.swift
            ├── Board.swift
            ├── Piece.swift
            ├── Rook.swift
            ├── Michael.swift          -- AI easy level
            ├── Alex.swift             -- AI medium level
            ├── TomHuynh.swift         -- AI hard level
            ...
    └── Data                         -- Storage implementation with FileManager
        ├── FileStoreManager.swift
        ├── ModelData.swift          -- Application data model
        └── SessionManager.swift     -- Manage user session login/register
    └── Localizations                -- English/Vietnamese translation
        └── Localization.swift
    └── APIs
    └── Utils                        -- Background music and sound effects
        └── AudioManager.swift
    ...
    └── Fonts
    └── Musics
    └── Animations
    └── Modifiers
    └── Extensions
```

Appendix 1. Project structure.

Badge name	Condition	Badge Image
<i>Welcome first game</i>	Complete a game, does not matter win or lose	
<i>Seasoned Player</i>	Win at least 10 game, does not matter AI or 2 player smode	
<i>Strategist Savant</i>	Do at least one promotion move in a completed game	
<i>En Passant Tactician</i>	Do at least one en passant move in a completed game	
<i>Castle Commander</i>	Do at least one castling move in a completed game	
<i>AI Beginner Beater</i>	Win at least one game vs easy level AI	

<i>AI Challenger</i>	Win at least one game vs medium level AI	
<i>AI Conqueror</i>	Win at least one game vs hard level AI	

Appendix 2. Badges system.