

Aleksandar Ivanović - projects portfolio

Kasparovchess:

- <https://apps.apple.com/us/app/kasparovchess/id1533398101>
- Devices: iPhone (portrait)
- Worked in a team with [Vladimir Lukić](#)
- Timeframe:
 - September 2019 - February 2021 - development phase
 - February 2021 - April 2021 - QA phase + launch
 - April 2021+ - maintenance + small feature improvements
- Main description and app functionality:
 - Chess platform by Garry Kasparov - learn, train and play chess
- App architecture
 - MVVM - App content
 - MVC - Gameplay section
- Main Features:
 - The app itself is a combination of video platform for chess learning, with chess gameplay (vs AI, vs Random opponent, vs Friend, Correspondence games, Puzzles, Analysis and more).
 - 120+ screens with multiple states
 - For AI to work, we needed some magic to happen because the engine is written in c++ which is not compatible with Swift, so architecture in layers are:
SwiftUI>Swift>ObjectiveC>ObjectiveC++>C++
 - So, all UI, except for the gameplay is written in SwiftUI, Networking and app logic in Swift, Complex navigation in ObjectiveC, Game interface and logic in ObjectiveC++, and game engine is taken from open source Stockfish 10 that is written in c++
 - With this project kicking in, the SwiftUI was released from beta, so I started using this framework from day one.
 - Multiplayer gaming mechanism is implemented via WebSockets
- Splash screen with application loading
- Intro pages
- Onboarding process
 - Signup and login via email, google, facebook, apple, or skip (guest user)
- Main tab menu containing
 - Home, Playhub, Watchhub, Learnhub and Global ranking
 - All are lists that are configured from backend (Strapi is used), with custom and different option - opening different sections of the application
- My profile
 - Informations, rating, upload avatar
 - Statistics - cumulative + detailed with filter and graph
 - Progress - puzzle, lessons and masterclass

- Game history - list, filter and details
- Profile settings
 - Edit profile informations
 - Membership - implemented as InApp purchase Auto Renewal subscription
 - Privacy, blocked users, friends and notification settings
- Game settings
 - Move method - Tap & Drag
 - Boards and pieces look selection - with instant change appliance within applications
 - Gameplay settings
- Payments
 - Auto Renewal - Payment through AppstoreConnect - with connection on all platforms (users can purchase premium membership on web and android app also - and it is applied within the application)
 - 3 Roles - Guest, registered user and premium user with different access to the application features
 - T-Mobile coupons
- Notification screen - place for unread notifications
- In-App notifications (challenge requests, rematch, correspondence moves, postgame analysis finished, auto renewal subscription expiration, etc ...)
- Search options - search for users, articles, videos, lessons and masterclasses
- Main menu
 - Shortcuts, social media links (opened within SafariController), T&C, Privacy policy, newsletter subscription and contact
- Videos
 - Featured videos within tab menu sections
 - All Videos list with infinite load and filter options
 - Video sections implemented - Chess Journeys, Documentaries and Interviews
 - Video details - Dailymotion player implemented + video description
- Lessons
 - Main lesson list (800 + lessons) with infinite load and filter
 - Lesson details - similar to video details with additional sections:
 - Lessons in same chapter
 - Exercises (puzzle list within lesson)
 - From the same teacher and on the same topic
- Gary Kasparov Masterclass
 - Main list of all chapters + single chapter list
 - Masterclass Details with Dailymotion video player + description + related in chapter + progress tracking
 - Premium feature only
- Articles - featured, list and details (content handled from customized JSON)
 - Support for bold text, italic, list, titles, images, quotes and links
 - Article informations, author informations, related articles
- GCT (Grand Chess Tour) livestream section with WKWebView support for loading live board of games currently playing/played
- Play now - Gameplay options section
 - Quick play - Set time controls and more options (private/public) and search for players. Random pairing or search for online players and send challenge
 - Play with friend - same as Quick Play options but instead of random opponents, challenge recent opponents or users that are added as friends

- Play the computer - play against the Stockfish AI, set time controls, difficulty level and player side (black / white / random)
- Gameplay section
 - Game info in header (gradient color is changing based on user ratings)
 - Everything that chess experience is about. Play chess by the rules controlled by Stockfish engine. VS another opponent (multiplayer) or against the “machine”
 - Move highlighting with options for available moves for selected piece + premove option
 - Users info (name, country, avatar, rating)
 - User connectivity
 - Captured material advantage
 - Players time remaining
 - List of played moves
 - Tab button actions
 - Resign from game
 - Offer draw to opponent
 - Chat
 - Send reactions
 - Move back and forward through played moves (handled all edge cases where user can receive opponent moves while browsing through game)
 - Three dots menu - add as friend, block, report...
 - Reconnection ability when entering from background
- Postgame screen
 - Shown when the game ends - by checkmate, stalemate, draw accepted, user resign, timeout, connection lost...
 - Showing if player has won or lost (or drawn), user info, player score, rating change ○ Options to exit game, rematch the opponent, send “applause” or request for Post Game analysis (premium feature only)
- Postgame analysis
 - Premium feature only
 - In order to have PGA, users are sending requests to the backend, and when analysis is completed, users are notified via notifications.
 - PGA screen is containing:
 - Users info
 - Number of good / inaccurate / blunder moves within a game
 - Board showing the game and moves played
 - Description section per move
 - Fork - analysis of current move, with 3 next best moves + showing the alternative if user move wasn't good
 - Advantage graph - showing the user strength throughout the game
 - Tab button actions
 - Switch between fork and graph
 - Move through the game
- Puzzles
 - 8 daily puzzles every day
 - Create a puzzle playlist (from 50.000+ puzzles) - choose difficulty level and theme and start playing the puzzle train. Unlimited for premium users - limited to 20 per day for registered users. Guest user is not allowed to play
 - Puzzle gameplay

- Header - for lesson puzzle is showing difficulty level
- Move list - populating while playing correct moves
- Board with starting position based on FEN received from backend
- Moving pieces only if correct moves. For incorrect, move is reseted, stats are updated and user can try again
- After user has played correct move, AI is playing next based on solution array received from backend
- Puzzle statistics - cumulative win percentage, users puzzle rating + rating change and win/loss ratio. For exercises (puzzles within lessons) here is shown puzzle title
- Move info - side to play, correct move, incorrect move, and option to show the solution
- Tab buttons actions
 - Go back and forward through played moves
 - Show hint
- Correspondence chess - type of gameplay where users have 1+ day to play a single move. Opponents are notified via notifications when the opponent has played the move, or when the game is finished. Also, you can find all correspondence games within Playhub section
- Global ranking - Rank list of all users on the platform, with filtering options like country and game type
- Firebase analytics + crashlytics monitoring
- Planned future features
 - Chess programs
 - Live games spectate
 - Tournaments
 - Classrooms
 - Anti cheat detection
- Stats

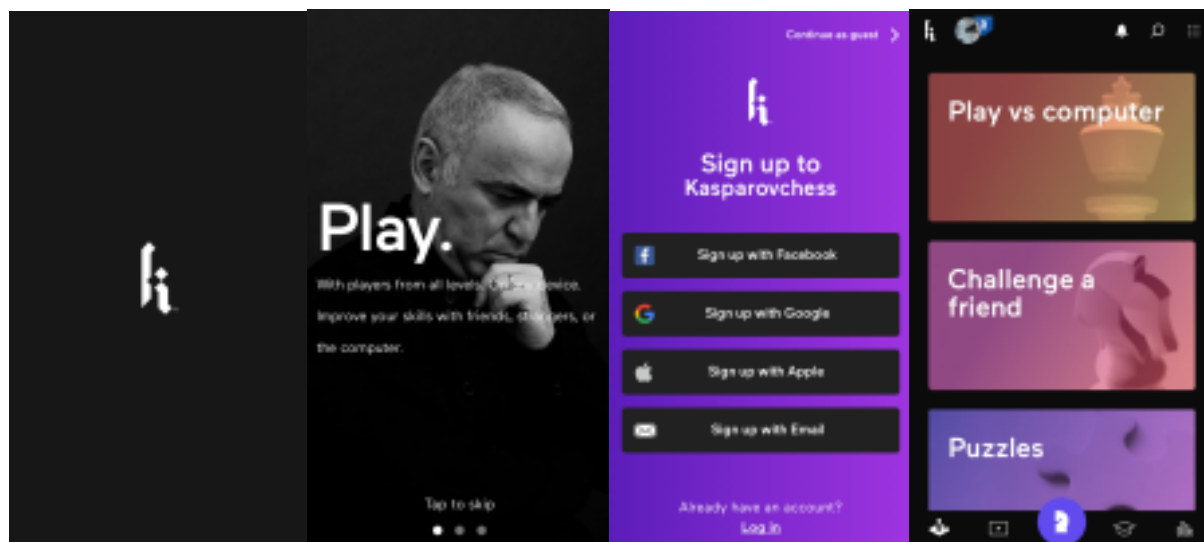
```
[Vladimirs-MBP-2:KeySquareChess_iOS sydbassetts$ cloc . --exclude-dir=Pods
2146 text files.
2011 unique files.
174 files ignored.

github.com/AlDanial/cloc v 1.90 T=2.33 s (847.6 files/s, 94387.7 lines/s)
-----
Language               files      blank      comment      code
-----
Swift                   808        24505        30554        74477
JSON                    856         0           0        19461
XML                      86          8           1        17413
Objective-C++           57         4782        5682        13916
C++                     34         3398        2817        10117
C/C++ Header           111        2224        2284         4909
Objective-C             14         691         770         1539
JavaScript              1          28          38           97
HTML                    1           5           0           57
CSS                     3           7           0           40
Lua                     2           0           4            8
Markdown                1           1           0            1
-----
SUM:                   1974       35649       42150       142035
-----
Vladimirs-MBP-2:KeySquareChess_iOS sydbassetts$
```

- Screenshots

- All screenshots are on the link:

<https://drive.google.com/drive/folders/1epG5yeTvTgOzdHiV-ehr-4rAfyHxtDgL?usp=sharing>



NovinarnicaPlus:

- <https://apps.apple.com/us/app/novinarnica-plus/id1554147691>
- Devices: iPhone (portrait) & iPad (portrait - landscape is planned for future improvements)
- Main description and app functionality:
 - Digital platform to read all newspapers and a lot of books on Serbian and other Balkans languages.
- App architecture
 - MVVM - App content
 - MVVC - Readers (inherited)
- Main Features:
 - The app is written fully in SwiftUI (except for reader that are inherited - ObjC) and networking layer is written in Combine (which made the app lightening fast)
- Splash screen with application loading
- Books and Newspapers lists (two column for iPhone, three for iPad) + categories separately
 - Home - Promo magazines, free books, free newspapers
 - Newspapers/Books - New, Recommended, Categories
- Search - both newspapers and books, browse and search by publishers
- Details
 - Cover image (with download in progress overlay), description, add to library, similar list of items
 - Action button
 - States - download, cancel, read, login (opening modal), coming soon
 - Books specific
 - Impressum with link to publisher, from same author
 - Newspapers specific
 - Archive with date pickers, notify when new issue is released, newspaper addons (not all newspapers have it)
- Readers - inherited from ancestor, with refined UI to be applied with new design
 - Newspapers
 - Download, unpack with removing protection from the the zip file of newspapers containing the array of images, thumbs and metadata JSON
 - Swipe to change current page (with indicator index)
 - Zoom in and out to focus the articles that user wants
 - Menu for fast going through pages in horizontal scroll
 - Books
 - Handle huge downloaded .html file in WKWebView
 - Customised css and javascript functions for handling the content
 - Book Content list with clickable items that links the selected chapter
 - Change Text size option
 - Bookmark when saved, or on back action
 - Adapts to Day or Night mode
- Login with username and password - no registration.
 - Registration is enabled on website and android
 - Multiple device restriction (max 2)
- Account
 - Service status
 - Free trial, active, inactive

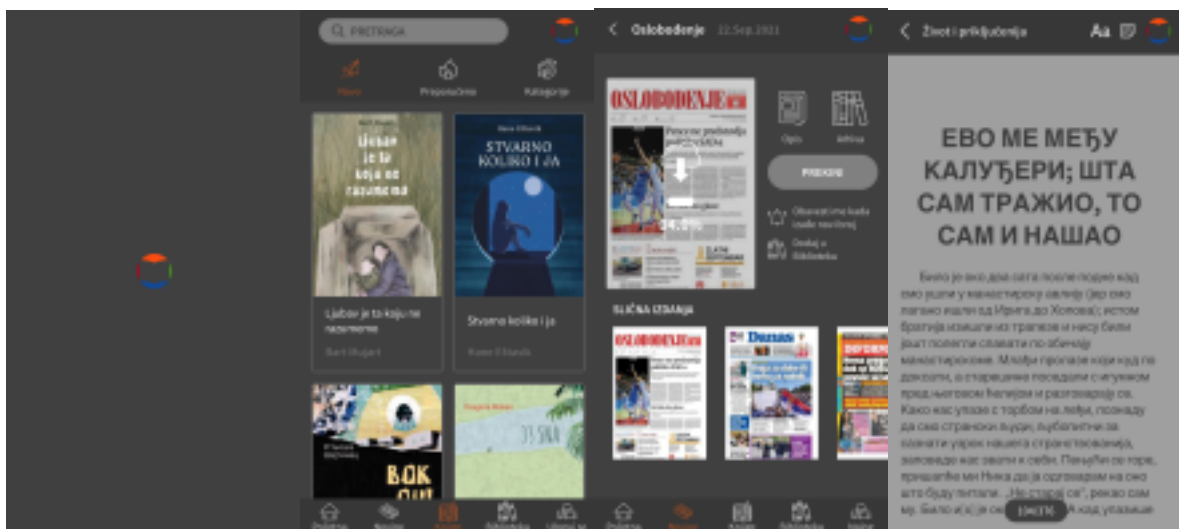
- Change password, logout
 - Day / Night mode - instantly changed and applied to all app components
 - Static screens - T&C, help contact
- News
 - Lists divided by categories
 - Details with header, cover image, content and link to origin newspaper issue
- Library
 - Downloaded (max 20 books, max 20 newspapers). If limit is reached, showing quick delete modal with option of multiple items deletion for freeing up device space fast ○ Delete option for downloaded, or removing an item from library
- Offline
 - Ability to read all downloaded books and newspapers
 - Option to reconnect
- Stats

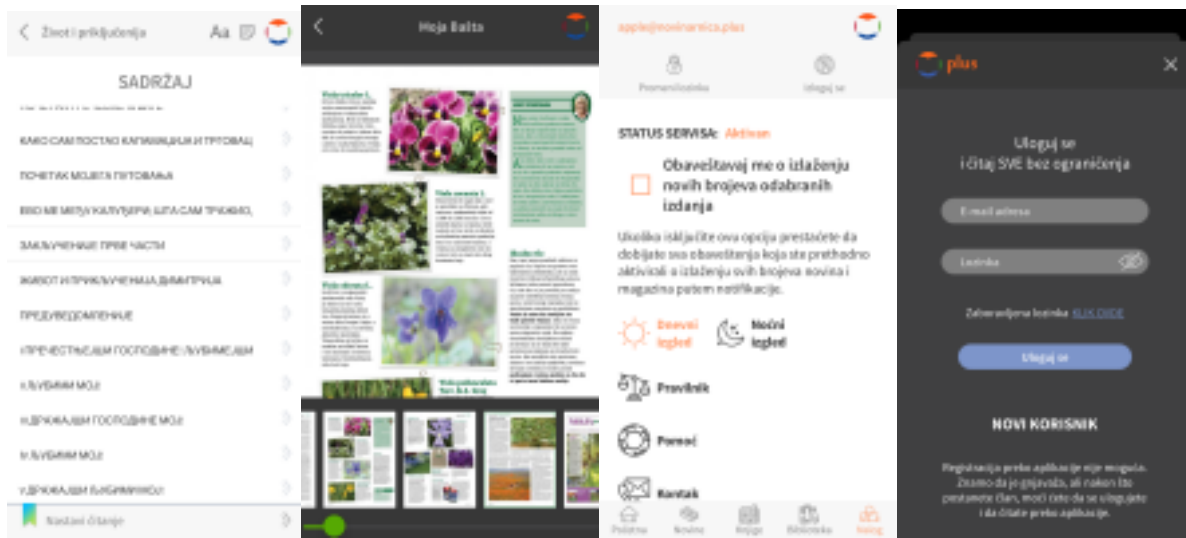
```
(Vladimirs-MBP-2:NovinarnicaPlus-iOS-new sydbassetts$ cloc . --exclude-dir=Pods
466 text files.
449 unique files.
29 files ignored.

github.com/AlDanial/cloc v 1.90 T=0.56 s (781.5 files/s, 98512.1 lines/s)
-----
Language               files            blank           comment           code
-----
Swift                   313              8784             9097             23793
XML                     20                0                10              8227
JSON                    80                0                 0             1888
Objective-C             12               627              807             1629
C/C++ Header           13              130              131              210
Markdown                1                 1                 0                 4
-----
SUM:                    439             9542            10045            35751
-----

Vladimirs-MBP-2:NovinarnicaPlus-iOS-new sydbassetts$
```

- Screenshots
 - All screenshots are on the link:
<https://drive.google.com/drive/folders/1epG5yeTvTgOzdHiV-ehr-4rAfyHxtDgL?usp=sharing>





Scratchy(removed from app store):

- Video link available at: <https://drive.google.com/file/d/17ttXv0XkbeuWpSPJMfKlQ3h973X7ffh/view?usp=sharing>
- Devices: iPhone (portrait)
- Main description and app functionality:
 - Scratching single or multiple cards to win various brands offerings, iterating over the scratching fields in vertical world map.
- Main Features:
 - Intensive use of UIKit. Custom vertical ScrollView world.
 - Complex UI all over the app
 - Referrals system, prizes lists, scratching cards limits and boundaries...
 - Numerous API calls handled with GCD(Background, main, concurrent queues)
 - Social logins, Firebase analytics..

Gun.deals (SlickGuns):

- <https://itunes.apple.com/us/app/slickguns/id1078583897?mt=8>
- Over 2 million downloads from Apple AppStore
- Devices: iPhone (portrait), iPad (portrait, landscape)
- Main description and app functionality:
 - Gun Deal Sharing community and Gun Search Engine in USA
- Main Features:
 - Different layouts (design implementation) on iPad and iPhone
 - Complex main menu
 - Navigation through different lists (genres, categories, rating, custom search)
 - Guns Gallery
 - Commenting (only reading for now)

MTVTR:

- <https://apps.apple.com/us/app/mtvtr/id1473449966>
- Devices: iPhone (portrait)
- Main description and app functionality:
 - Fitness platform for exercising
- Main Features:
 - Displaying bundle of exercises users need to do with a timer and statistics
 - Chat, followers list, live forum
 - Many types of Push notifications
 - Alamofire with Moya and Promises used for networking layer
 - Complex PageViewController

MeilleureSCPI:

- <https://apps.apple.com/lk/app/scpi/id721721575>
- Devices: iPhone (portrait)
- Main description and app functionality:
 - Platform for finding real estates nearby(in the radius of 2kms of the user) in France, Paris.
- Main Features:
 - App translated from Objective C to Swift fully
 - Local notifications logic implemented. When user passes by specific real estate in Paris, local notification is displayed with real estate basic info. When clicked on notification, it leads user to real estate details screen.
 - Apple Maps

Samo Ne Autom:

- App removed from the App Store
- Devices: iPhone (portrait)
- Main description and app functionality:
 - Tracking and drawing live polylines for users movement activity on the map
- Main Features:
 - Apple Maps, Drawing polylines live, calculating time and distance user covered
 - Tracking user activity offline also, with Realm database. Syncing data with the server
 - Lists with user rankings, as everything was part of a tournament

* NOTES:

- I was also involved in many other projects, i didn't mention them above because those were either small projects, or I was working on some crash fixes, some small tweaks, bugs finding and design changes. Or those projects are not on app store anymore.
- For the great majority of those apps above, I have been working solo on them (more than 95%, excluding the project Kasparovchess), from File->New Project to the appstoreconnect.apple.com.