Switch Shoutout Development Log

2/14/18

* Created project
* Created basic launchscreen
* Added firebase cocoapods
  + Firebase
  + Firebase/Auth
  + Firebase/Database
* Created welcome, register, login, and home view controllers
* Linked segues from welcome buttons to registration/login view controllers
* Linked segues from registration/login view controllers to home view controller
* Integrated firebase registration and authentication for registration and login view controllers respectively
* Added navigation items to each view controller for a title
* //Added a sign out button to home view controller hooked up to firebase
* Hid failed login/registration labels using viewWillAppear
* Added a model for a Game object
* Added a friend code field to the registration screen
  + It gets added to the “users” table under the uid of the user
* Added home, my games, friends, and settings views
* Added a tab bar controller to point to the home, my games, friends, and settings views

2/15/18

* Created custom cell for My Games with UILabel and UIImageView
* Made cells able to be deleted
* ~~Removed navigation bar from tab bar controller via self.navigationController?.setNavigationBarHidden(true, animated: false) in HomeViewController’s viewWillAppear~~
  + ~~When the navigation bar was present, it kept additional bar items from appearing, namely an Add button for adding games and friends in their respective view controllers~~
  + ~~Needed to embed each tab bar sub-view controller in their own navigation controllers~~
* Embeded each tab bar sub-view controller into their own navigation controllers
* Added plus sign button to right side of navigation bar in My Games and Friends view controllers
* Set the Settings view controller content type to Static in the attributes inspector
* Implemented a sign out cell in the settings table view controller
  + Upon touch up, pops up an alert asking if the user is sure
    - If user touches ‘No’, alert is dismissed with no further action
    - If user touches ‘Yes’, an attempt to log the user out is made to firebase
      * ~~Upon a successful logout, the user is brought back to the login screen~~
      * If the sign out attempt fails, an alert appears letting the user know. No further action is taken
* ~~Fixed a bug where navigation bar would not reappear when popping back to the welcome screen~~
  + ~~Needed to show navigation bar under viewDidAppear rather than viewDidLoad~~

2/16/18

* Navigation overhaul
  + Unlinked login navigation stack from tab bar navigation stack
  + Added check in app delegate didFinishLaunchingWithOptions to see if user is logged in
    - If not, set rootViewController to welcome screen
    - If so, do nothing and start at tab view controller as normal
  + Added UIWindowTransitions pod to project (<https://github.com/malcommac/UIWindowTransitions)>
    - Allows use of UIApplication.shared.keyWindow?.setRootViewController(viewController) to create an animated transition
      * Using rootViewController (as in app delegate above) transitions with no animation, so this allows for a push-like transition without the need for segues or additional navigation controllers
    - These transitions are applied when the user logs in, registers, or signs out
      * Log in and registration transitions to tab bar controller, while signing out transitions to the welcome screen
* Made email address text field on log in screen and registration screen the first responder
  + Changed keyboard to email
* Changed settings tableview to have 3 sections
  + Made email cell a label
    - Made cell not selectable
  + Made friend code field a label that has a disclosure indicator which is linked to the Change Friend Code table view controller
* Created ChangeFriendCodeTableViewController
  + Has 1 cell to enter new friend code
    - Cell contains textbox to enter new friend code
      * Made textbox is first responder
  + Added save button on right side of navigation bar that update the friend code for the current user in firebase
    - Button pops current view controller and shows settings view controller
* Created array of game objects in app delegate to save to core data on first launch
* Made Game class conform to NSCoding
* Encoded gameList to NSData and synchronized to NSKeyedArchiver in app delegate
* Created xib for shared cell between MyGameTableViewController and AddGameTableViewController
  + Registered xib with each tableview (mygame and add game)
* In AddGameTableViewController cellForRowAt, set each table cell’s image and label to the respective game’s image and title
* In MyGamesTableViewController and AddGamesTableViewController, set table view cell row height to 80 to match the GameTableViewCell custom cell
* Disabled selection for MyGamesTableViewController
* Created a toggle for checkmarks in cells in AddGameTableViewController

Issues:

* ~~MyGamesTableViewController only updates after logout/login or when restarting the app, even though the game list should be updated in viewWillAppear~~
* ~~AddGamesTableViewController cells are retaining the checkmark when the cells are being recycled~~

2/17/18

* Saved checkmarked items from AddGameTableViewController to be presented in MyGamesTableViewController
  + Added a new property to Game called ‘added’ (bool)
    - added this property to the initial game list in app delegate
  + Pulled the game list in AddGameTableViewController; updated the added property of each game on viewWillDisappear
  + Pulled the game list in MyGamesTableViewController, created an array to hold only the games with added=true, and displayed that array
    - Needed to clear the array before repopulating it with an updated list from AddGameTableViewController
    - Then needed to run reloadData() on tableView
* Updated game list when games are deleted from myGamesTableViewController
  + Had to search iteratively through both gameList (user deaults array) and myGamesList (local to class) using indexed for loop. Couldn’t use indexPath.row as index because games change rows when rows get deleted, so indexPath.row couldn’t be predicted/accurate
  + Saved game list in viewWillDisappear
* Added ‘else’ to if statement regarding a game’s .added bool value and its checkmark status
* Implemented pod ‘InputMask’ to allow for custom formatting of the friendcodeTextfield in the ChangeFriendCodeTableViewController
  + Creates a mask of SW-####-####-####
* Changed the tableViewController to just a label and textfield

2/18/18

* Created Beacon class
* Added ‘Create Beacon’ button to home tab
* Created ‘ChooseAGameTableViewController’
  + This closely resembles AddGamesTableViewController, but functionally it will return to the home screen after a cell is selected and the user taps ‘Done’

2/19/18

* May not need a Beacon class since the game title can be stored textually in Firebase and the beacon will be stored under each user’s table
  + Will leave Beacon class alone for now; will remove later if deemed unnecessary
* ~~Created protocol ‘ChoseAGameDelegate’ that has a method ‘userChoseAGame’~~
  + ~~Passed beacon game title to HomeViewController via delegate in didSelectRowAt in ChooseAGameTableViewController~~
* ~~Made HomeViewController conform to ChoseAGameDelegate and implemented func userChoseAGame~~
* Don’t need a protocol/delegate since the database can be updated straight from the ChooseAGameTableViewController
  + Implemented this functionality in ChooseAGameTableViewController
* Added labels to HomeViewController to hold the beacon game title for the current user
  + Added database call to viewDidLoad to fetch current beacon name; if none, label.text = “None”
* Hid tab bar on ChooseAGameTableViewController
* Removed back button from AddGameTableViewController and added Done button for UX clarity
* Added check on Create Beacon button press for existing beacon
  + If the current user has an active beacon, an alert will appear telling them that they cannot have more than 1 active beacon and will prevent the user from navigating to ChooseAGameTableViewController
* Created Remove button for beacon on HomeViewController
  + Will appear only if user has an active beacon
  + Will set the beacon database value to nil if pressed and set the game label text to “None”
* Added game images to firebase storage
* Added games to firebase under games/
  + Each game is a dictionary with the title as the key and the path to its image as the value
* Started reworking AppDelegate’s populating of the game array to pull game info from Firebase instead of having it pre-loaded in the app

2/20/18

* ~~Successfully reworked AppDelegate’s populating of the game array to pull from Firebase~~
  + ~~Side affects:~~
    - ~~Currently the .added bool is set to false upon every app startup~~
    - ~~When switching to My Games or the AddGamesTableViewController, there is a bit of lag~~
* Added if block in AppDelegate to check if the games list is larger than the previously recorded size
  + Stores games table childrenCount in User Defaults
  + If the counts are the same, it will skip re-downloading the new data
  + If the counts are different, it will download all of the game data
* Cleaned up AppDelegate code with more functions to organize blocks of functionality
  + Try to undo this. This is what caused loading games to break.

2/21/18

* Reverted to older version of AppDelegate.swift and was able to restore functionality
  + Saved a copy in Switch Shoutout root directory named AppDelegate\_working\_copy\_2\_21\_18.swift
* Reverted to non-functioning copy from the night of 2/20/18 to apply fix
  + Fix was to load the gameArray into userdefaults inside of the pullGameImages completion handler
    - Wrapped this block in an if statement checking if we were at the end of iterating through finalGameDictionary so we aren’t setting the userdefaults every iteration
* Fixed a bug where the set of new games being downloaded overwrote the existing set of games
* Hid the Tab Bar on AddGameTableViewController
* Added icons for tab bar
* In MyGamesTableViewController’s viewWillDisappear, enclosed the saving to NSKeyedArchiver code in an async DispatchQueue to get rid of the hang on the main thread when navigating away to another tab

2/22/18

* Added functionality to remove games from Firebase
* Started adding functionality to remove beacon if it matches a game that got removed from firebase
* Commented AppDelegate.swift
* Disabled the done button on ChooseAGameTableViewController until a cell is selected
* Completed functionality to remove beacon of game that was deleted from firebase

2/23/18

* Moved all code for checking for new or removed games to HomeViewController’s viewDidLoad
  + This needed to be done because there was no authentication to fetch the images if the user was not logged in
* In HomeViewController, moved viewWillAppear’s check for existing beacon code into helper method checkForBeacon()
* Set the remove button’s isHidden status to true if the app is on its first launch
* Fixed a bug in pullGameDictionary() where the completionHandler would be called twice if the user had an active beacon, causing the games to be loaded twice
* Fixed a bug where Auth.auth().currentUser.uid was being assigned to a global variable, which is processed before FirebaseApp.configure() is ran
* Changes made to My Games when using Add Games are now reflected in Firebase under the “games” child of the user
* Added function setMyGames() in HomeViewController to pull users’ My Games lists from Firebase
* Changed the label properties on ChangeFriendCodeViewController to better fit the text for entering an invalid friend code
* Added an alert blocking the user from adding a beacon if they have no games in My Games
* Added functionality to remove a user’s game from Firebase if they delete it from My Games
* Fixed a bug that caused the Add Games list to not update
  + Changed the My Games viewWillDisappear function to only be asynchronous if the edit button did not cause the transition

2/25/18

* Made save button on ChangeFriendCodeViewController enabled only when the appropriate number of characters have been filled in the text box
  + Removed the label specifying friend code format since the user is locked into the format through InputMask
* When transitioning from MyGamesTableViewController to AddGamesTableViewController, used prepareForSegue to transfer game list rather than NSKeyedArchiver
  + This drastically cuts down on load time between the two view controllers
* When transitioning from AddGamesTableViewController to MyGamesTableViewController, used AddGamesDelegate method passDataBack to transfer game list rather than NSKeyedArchiver
  + This drastically cuts down on load time between the two view controllers

2/27/18

* Moved My Game list loading logic to viewWillAppear from viewDidAppear
  + This cut out loading time, which makes SVProgressHUD unnecessary even though I basically added the loading time to use it
* Began sorting through code after creating a myGameList in NSKeyedArchiver to be separate from sortedGameList
  + The goal of this is to cut down on loading and rewriting the sortedGameList
  + Leaving a bug where the wrong list is being passed from My Games to Add Games

2/28/18

* Changed all instances of accessing/saving “sortedGameList” to either “MyGamesList” or “AllGamesList”
* Removed the added property from class Game
* Removed all references to the added property of class Game and replaced with a nifty function to filter out an array with a given condition
  + Ex. let gameArray = passedGameList.filter {$0.title == gameTitle}
* Finished implementation of NSKeyedArchiver for “AllGamesList” and “MyGamesList”
  + No lag or loading when switching between tabs or from My Games to Add Games and vice versa
* Set the beacon Remove button to hidden in HomeViewController’s viewWillAppear
* Removed the beacon if it matches a game being deleted from My Games or Add Games
* Fixed a bug where all games would be duplicated when a new game was added or removed in Firebase

3/1/18

* Fixed a bug where a user’s beacon would be deleted from Firebase when they changed their friend code
* Added functionality to remove a game from My Games, the user’s games in Firebase, and the user’s associated beacons in Firebase (if applicable) when a game is removed from Firebase
* Created ChangeNicknameViewController
  + Copied from ChangeFriendCodeViewController
* Added nickname cell and textfield to SettingsViewController
* Created segue with identifier “goToChangeNickname”
* Added functionality to make save button in ChangeNicknameViewController to only enable when there is text in the textfield
* Changed Add Friend view controller into a tableview controller
* Created AddFriendTableViewController.swift
* Process of implementing a search bar in AddFriendTableViewController:
  + Created extension of AddFriendTableViewController and made it conform to UISearchResultsUpdating
  + Declared UISearchController variable
  + Set searchController.searchResultsUpdater = self
    - Basically the delegate for the search bar
  + Set searchController.obscuresBackgroundDuringPresentation = false
    - Since we are not using a different view to display the results, we do not want our view to be obscured
  + Set searchController.searchBar.placeholder = "Search for friend"
  + Set navigationItem.searchController = searchController
    - Must do this outside of IB because UISearchController isn’t supported yet
  + Set definesPresentationContext = true
    - Hides the searchbar if the user navigates to another view controller while the UISearchController is active
  + Grabbed list of all users in viewWillAppear
* Have somewhat functioning search bar

3/2/18

* Added InputMask to friendcode textfield in registration view
* Removed the “SW-“ label from the registration view
* Added checks on the registration submit button to check for completed friend code and nickname fields
* Decided to axe the searchController in favor of a textfield and search button
  + Don’t want users to see users they aren’t looking for
* Added text fields to search by nickname or friend code in AddFriendViewController
* Made AddFriendViewController conform to UITextFieldDelegate
* Implemented textFieldShouldReturn(…) to provide functionality when the user presses the search button on the keyboard for the nickname textfield
* Added separate search functions for nickname and friend code
* Added InputMask functionality to friend code textfield
* Implemented code from <https://stackoverflow.com/questions/28338981/how-to-add-done-button-to-numpad-in-ios-8-using-swift> to add a custom UIToolbar on top of the friend code textfield number pad keyboard
  + The toolbar holds a search button that gets enabled when the input mask is filled
* Overrode prepareForSegue to pass search data to SearchResultsTableViewController
* Created FriendRequestTableViewController and corresponding swift file

3/5/18

* Created UserTableViewCell.swift and UserTableViewCell.xib for use with tables of users
* Added button to UserTableViewCell.xib for add/confirm purposes
  + Game button rounded corners in User Defined Runtime Attributes

3/6/18

* Added functionality to SearchResultsTableViewController’s addButtonPressed to create a friend request key/value pair in the requestee’s user table in firebase
  + Key/value pair is “uid”: current user id

3/7/18

* Completed fried request system
  + When a request is sent, a uid/“true” key/value pair gets created under the request receiver’s friend\_requests key
    - This has two key/value children:
      * “nickname”/nickname
      * “friend\_code”/friend code
  + The friend request list of the request recipient is populated with the nickname and friend code of the requestor
  + When the confirm button is pressed, the friend request, along with its children, is removed and a key/value pair is created in the recipient’s friends table
    - “friend”/uid
* Edited filling of allUsers[] in viewDidLoad in AddFriendViewController to use less database calls by comparing to the user’s uid instead of friend code
* Added functionality to change the search results cell button’s properties depending on friend request status:
  + Not added or requested: Blue add button
  + Requested but not added: Gray request sent button (not interactable)
  + Added: Gray added button (not interactable)

4/5/18

* Essentially loaded friends into FriendsTableViewController
  + Need to create user defaults storage for friends to avoid waiting on closures every time the view controller loads
* Saved friends to user defaults in homeviewcontroller
  + Friends list is now correctly populated
* Removed FriendList from user defaults upon sign out

4/8/18

* Added functionality to remove friends
* Added tableview.reloadData() to populate friends table after returning from friend requests view
* Disabled “Confirmed” friend request button interaction
* Created empty friend list for instance of friend being deleted by other user when user only had one friend

4/10/18

* Got rid of implementation of saving friends to user defaults in order to refresh the friends list on viewWillAppear) by making a firebase call
  + This allows for removal of users from your friends list who have deleted you as a friend without having to close and re-open the app

4/13/18

* Set SVProgressHUD.setDefaultMaskType(.black) in viewDidLoad in HomeViewController
  + Creates a mask to disable user interaction when loading icon is present
* Fixed a bug where users who were added as friends were able to be added again
  + Now appropriately shows grey “added” button instead of blue “add” button
* Added ReachabilitySwift pod to handle loss of internet connection
  + Current implementation is in HomeViewController; look into putting it into tabbarviewcontroller for global access

4/24/18

* Moved reachability code to viewDidLoad from viewWillAppear in HomeViewController to reduce the amount of times it gets called
* Replicated reachability code to welcomeViewController

5/1/18

* Welcome Navigation UI Work
  + Created image for buttons
  + Created image for navigation bar back buttons
  + Created image for text field background
  + Created image for welcome screen
  + Changed font family to Avenir
  + Changed background color to red
  + Hid navigation bar on welcome screen
    - Used animated: true for smoother transition
      * Changed this for all occurrences of hiding or showing the navigation bar
* Changed the rest of the app’s textfields and labels to Avenir family
* Changed all navigation bar fonts to Avenir-heavy via app delegate
* Put the reachability code in its own function for cleanliness
* Made it so that either only the Create Beacon button is showing, or only the beacon labels and the remove button are showing
* Made the tab bar opaque so that it will look better with an image background
* On both FriendsViewController and AddFriendViewController, checked the box “Extend Edges Under Opaque Bars” to use an opaque bar for the HomeVeiwController and avoid issues with AddFriendViewController not showing the tab bar

5/2/18

* Set navigation bar background color to red and text to white
* Set status bar style to light for white text
  + Also had to set a property in the plist to disable view controller overriding of the status bar style
* Updated table cell background images
* Created new table cell image to replace checkmark accessory as it was horizontally shrinking the background image
  + Replaced functionality to add and remove checkmark with background images
* Changed “Request Sent” text to “Pending”
* Changed launch screen picture
* Created new table cell image for selected cell in ChooseAGameTableViewController
* Changed some images to a more appropriate resolution
* Changed the font for the UserTableViewCell
* Changed the color of the “Confirmed” friend request button to gray
* Alphabetized the My Games list in viewWillAppear to fix a bug where the list would occasionally become out of order
* Added animated tab bar transitions via <https://github.com/franklinsch/iOSScrollingTabBarAnimation>
* Removed code to hide tab bar in AddFriendTableViewController and AddGamesTableVieController and checked boxes for “Extend edges under opaque bars” and “Hide bottom bar on push” to better animate hiding of tab bar
* Made the navigation bar not translucent and made it appear on viewWillLoad for non-home tab bar view controllers (including choose a game view controller) to create better nav bar transition animation from home to other view controllers
  + Had to make unhide animated for choose a game view controller
* Checked boxes for “Extend edges under opaque bars” on login and register view controllers to fix layout issues from opaque nav bar

5/3/18

* Created tableView in HomeViewController
  + Conformed HomeViewController to UITableViewDelegate and UITableViewSource protocols
  + Changed the tableView cell height to 60
  + Disabled cell selection
  + Disabled Bounce on scroll
* Created BeaconTableViewCell class and xib
  + Added imageView and 2 labels for game picture, game name, and friend nickname
  + Gave it “beaconCell” reuse identifier
  + Registered xib with tableView in viewDidLoad
  + Changed the cell’s imageView content mode to Aspect Fit
* Added code to checkForBeacon() to check for friends’ beacons
* Moved SVProgressHUD.dismiss() to post-checking for friends’ beacons
* Placed a SVProgressHUD.show() in viewDidLoad to ease the transition back to the home screen
* Redesigned the home screen

5/4/18

* Added an alert in checkForBeacon if there is no data in AllGamesList that notifies of a possible slow connection
* Added the logo back to the home screen

5/5/18

* Fixed a bug where SVProgressHUD would dismiss early when transitioning back to the home screen from Friends or Settings
  + SVProgressHUD was being dismissed at the end of Firebase calls which carried over to the Home screen (on a slow connection)
* Added a TableViewCell in SettingsViewController that redownloads games
  + Added if blocks in HomeViewController for when SettingsViewController is calling its functions to not reference resources that aren’t on screen

5/6/18

* Fixed a bug causing selected cells in ChooseAGameViewController to be reused when scrolling

5/7/16

* Added DNZEmptyDataSet to MyGamesTableViewController and AddGamesTableViewController

5/9/18

* Changed the color scheme of GameTableViewCell, UserTableViewCell, and AddFriendsViewController

5/11/18

* In HomeViewController, when games were downloaded for the first time, games in My Games weren’t being sorted until navigating to MyGamesTableViewController
  + Inserted a function to sort them
* Darkened the text on the empty tableview controllers for My Games and All Games
* Showed the network loading indicator on the status bar when reloading games from SettingViewController

5/14/18

* Fixed a bug where My Games would be empty upon first load even if the user had added games before (race condition)
  + Needed to put a condition on SVProgressHUD.dismiss() in the tableView
* Fixed a bug where the user defaults wasn’t getting updated when dismissing the AddGamesTableViewController
* Removed the alert popup from redownloading games and changed it to a section header
  + The reachability pod doesn’t seem to work when an alert is being presented
* Included runReachability() in the HomeViewController instantiation when redownloading games

5/15/18

* Fixed a bug caused by re-running runReachability() after redownloading games that lead to a crash (last bullet point on 5/14)
  + Included runReachability() in the excluded if block for redownloading games in viewDidLoad in HomeViewController
* Set autolayout contraints for ReachabilityViewController

5/17/18

* Set autolayout constraints for WelcomeViewController

5/21/18

* Removed emailTextfied.becomeFirstResponder from RegisterViewController and LoginViewController
* Added autolatyout constraints to RegisterViewController and LoginViewController

5/22

* Added git repository for switch shoutout on bitbucket

5/24/18

* Added .gitignore
* Added autolayout constraints to AddFriendsViewController
* Added autolayout contstraints to SettingsViewController
* Added autolayout constraints to ChangeFriendCodeViewController
  + Still could use some adjustment for larger screens

5/25/18

* Used top constraint equal to Center Y with a multiplier < 1 for better autolayout constraint on ChangeFriendCodeViewController
* Set the navigation title for ChangeFriendCodeViewController to a label with a variable text size so that the text doesn’t cut off on the iphone se
* Set the cursor color (tint color) to black for text fields on ChangeFriendCodeViewController, ChangeNicknameViewController, and AddFriendsViewController
* Changed Home screen gray stripes from UIImageViews to UIViews
* Added constraints to BeaconTableViewCell
* Changed Create Beacon Button insets to help auto adjust the button title so it would be a smaller font size on the iphone se
* Added autolayout constraints to HomeViewController

6/5/18

* Added temporary genre images to firebase
* Set the Add button (to send a friend request) to have an adjustable title text size to prevent the text “Request Sent” from being clipped
  + Added insets so the text wouldn’t be too close to either side of the button
* Changed the bar button item text from “Friend Requests” to “Requests” to make it fit better on the iphone se

6/7/18

* Added @1x and @2x images for all image sets
* Created imageview to replace checked background for gametableviewcell
  + Allows for correct aspect ratio of checkmark circle on every device
* Added autolayout constraints for the rest of gametableviewcell
* Added back check in cellForRowAt to make sure the checkmarks don’t get reused on cells when scrolling
* Changed the separator color on friendrequesttableviewcontroller and searchresultstableviewcontroller to white
* Removed the beacon background image on the home screen
* Added DZNEmptyDataSet to FriendRequestsTableViewController and FriendsTableViewController
* Fixed a bug that would cause SVProgressHUD to not dismiss on the home screen if the user had added friends before but removed all of them and returned to the home screen

6/8/18

* Changed “Pending” friend request button text to “Request Sent” to match what it says after you press “Add”
* Modified layout of LoginViewController
  + Moved all elements in LoginViewController up to accommodate the keyboard
  + Made email text field first responder
  + Changed ‘Submit’ button text to “Log In”
  + Changed position/constraints of error message to appear in a better place
  + Made the Log In button disabled until both text fields had text
  + Added error handling cases for log in
* Split registration into 2 separate view controllers
  + View controller 1 has fields for email, password, and password verification
    - Upon pressing Register button, account is created and a segue to the next registration view controller is performed
  + View controller 2 has fields for friend code and nickname
    - Upon pressing Get Started, the friend code and nickname are uploaded to firebase and the user is segued to the home view controller
  + Added registration error handling
  + Made the buttons disabled until all fields have some text
* Made password2textfield a password field
* Removed back button from registration2viewcontroller to prevent the user from going back
* Fixed autolayout issues that caused the textfields in registration2 to not appear
* Made the friend code field conform to the appropriate input mask
* Added autolayout constraints to both registration view controllers

6/9/18

* Finished autolayout constraints on both registration view controllers
* Fixed ‘Get Started’ button enablement issues with the friend code field

6/13/18

* Added Chatto and ChattoAdditions pods
* Added new navigation controller and Conversations tableviewcontroller that connect to a 5th tab on the tabbarcontroller
* Created ChooseAFriendTableViewController
  + Displays a selectable friend list
  + Has cancel and done buttons
  + Updates the enabled state of the Done button based on whether or not any friends are selected

6/14/18

* Updated app icon (still needs work)
* Made Done button on ChooseAFriendTableViewController go to a chat window
* Added a back arrow left bar button item to the chat window view controller to navigate back to the conversations view
* Added checkmarks to the cells to visually represent whether or not they were selected
* Upon hitting done, creates a child in firebase under “conversations” with a unique id constructed from appending all other chat group members’ uids separated by an underscore (this may not be necessary with FCM)

6/15/18

* Renamed AddRandomMessagesChatViewController to ChatWindowViewController
* Added functionality to set the title of the chat window to the names of the users you’re chatting with
* Merged DemoChatViewController with ChatWindowViewController
* Added conversationUID as a variable that can be passed into ChatWindowViewController so it can send it along to the datasource
* Altered addTextMessage in DemoChatDataSource to have senderID (userUID in firebase) and conversationID (conversation autoID in firebase) as extra parameters so it can add messages to firebase
  + Populates messages under the conversation’s autoID in the conversations table
  + This reduces redundancy when compared to having the messages appear under each user’s table

6/21/18

* Removed ‘Demo’ from filenames and class names in the Chat View Controllers folders and updated the appropriate references
* Created ConversationTableViewCell swift file and xib

6/22/18

* Added timestamp to messages
* Added basic Conversation view functionality
* Added rudimentary functionality to open up existing conversations
  + Seems to load messages in random order as well as on a random side of the screen

6/23/18

* Got messages to appear on the correct side when opening an existing conversation
  + Messages still appear in random order and messages do not dynamically load while in chat window

6/25/18

* Added functionality to remove the conversation from the current user’s table when they remove it from the conversations table view
  + Currently, this removes the conversation from all members’ tables AND the main conversations table. This is enabled for testing purposes. The only time this should happen is if all members have removed the conversation from their respective user tables (conversations table view)
* Added functionality to remove a conversation from the conversations firebase table AND all members’ user tables if a new conversation was created, but the initiating user closed the chat window before any messages were sent
* Fixed a bug where the current user’s nickname was included in the chat title in the chat window and in the conversations table view cell

6/26/18

* Fixed a bug that would result in all members of a conversation not getting added to the conversation object by the time the cell was displayed
  + Moved the last message fetching request (which included the line to add the new conversation to the array) inside the members fetching request and made it so the table would only get reloaded if all members have been added and all conversations have been added
* Ran swift 4.2 conversion only on switch shoutout files
  + Ended up not working with pods; reverted
* Tweaked ConversationTableViewCell layout
* Somehow last message in conversationstableviewcell is updating correctly now
* Added functionality for date/time field in ConversationTableViewCell
  + If the last message was sent in the same day, the field displays the time it was sent
  + If the last message was sent yesterday, the field displays “yesterday”
  + If the last message was sent 1 week or less ago, the field displays “X days ago”
  + If the last message was sent over 1 week ago, the field displays the short date of when it was sent

6/27/18

* Fixed a bug causing SVProgressHUD to disappear early when loading the home screen
* Programmed the message status to begin with “sending” and move to a successful sent message
  + Messages will only show the “failed” status if the firebase setValue call returns with an error
  + If a user attempts to send a message while not on a network connection, the message will remain in the “sending” state until a connection is restored
    - Theoretically, this shouldn’t happen as the reachability popover screen will prevent the user from doing anything when there is no network connection
* Fix an issue where existing messages would be displayed out of order
  + Needed to sort through snapshot.children rather than snapshot.value
* Sorted conversations so that they will appear in order by time of last message received

6/29/18

* Added functionality that opens up an existing conversation if the user attempts to create a new conversation with the same members
* Showed SVProgressHUD when loading a conversation

7/2/18

* Denormalized database structure for conversations per <https://firebase.google.com/docs/database/ios/structure-data>
  + Split conversations table into three tables – conversations, members, and messages
  + Makes it so calls for fetching certain aspects of messages don’t require downloading the entire tree of conversations including messages
* Futzed around with some errors and realized they were coming from the listener in the table view – need to add a new conversation to the array

7/3/18

* Added observer in ChatWindowViewController to listen for new incoming messages
* Fixed a bug where the user’s sent messages were being received again as if sent by another user
* Followed comments on <https://github.com/badoo/Chatto/issues/148> to get a text-only input bar
  + Had to do a lot of auto layout finagling to keep the existing constraints due to them needing to be there for the parent functions

7/5/18

* Added an observer to listen for new messages on existing conversations and reload all data
  + Really should only update the one cell and move it to the top
* Revised the above functionality to add an observer in cellForRowAt and made it only edit the conversation item in conversationsArray and then reload the table to avoid pulling in all of the existing conversation data every time a new message is received

7/6/18

* Finished observer query to listen/add new conversations
* Fixed a bug where the conversations table was attempting to reload before the conversationsArray was repopulated
  + Used a temp array to store the new info in so that conversationsArray never has to be emptied; temp array data gets transferred to conversationsArray when it has finished populating
* Implemented a read system that marks unread message cells
  + Accounted for user sending own message, user receiving message, user receiving new conversation message, and user being on conversation screen of new message upon arrival
* Created alphabet avatars and added them to the assets folder

7/9/18

* Created avatar view controller that displays the alphabetical avatars to choose from
* Programmed the highlighting of a chosen avatar on AvatarViewController
* Added a cell to SettingsViewController to allow the user to choose a new avatar
* Created large alphabetical avatars

7/12/18

* Created Message class to handle storing message data when populating messages to be sent to ChatMessageFactory
* Created function updateConversations() in HomeViewController called in viewDidLoad()
  + Fetches conversation data for current user and stores it in user defaults
    - Does NOT store message data (would be too big and slow)
* Combination of the two above points led to solution for correctly displaying avatar images in ChatWindowViewController
* Fixed bug where text bubble tail and avatar behaved as if every conversation was a 2 person conversation – all incoming text messages were treated as if coming from the same person
* Added sender firebase id variable to CustomTextMessageModel and changed the type of the ‘currentMessage’ variable from MessageModelProtocol to CustomTextMessageModel

7/22/18

* Added functionality to pull conversation table cell information from user defaults for faster table loading
  + Messages are still being pulled from firebase
* Fixed SVProgressHUD dismiss issues for first load, normal load, and navigation cases for HomeViewController

7/27/18

* Updated cocoapods
* Removed the actionsheet alert from settings view controller when choosing an avatar in settings
* Added choose avatar button to registration2viewcontroller
* Added constraints to avatar icon in settingstableviewcontroller
* Added “ConversationsList” and “UserAvatar” to list of user defaults stores to remove on logout
* Completed functionality to choose avatar during registration
  + Added logic to check if avatar had yet been chosen to be an additional factor in enabling the Get Started button

7/30/18

* Changed firebase calls that pulled list of all users to search for specific users

9/6/18

* Updated cocoapods

9/8/18

* Updated last message in user defaults to show updated conversations list when pulling from user defaults

9/9/18

* Updated last message time in user defaults to show updated conversations list when pulling from user defaults
* Decided to scrap chat and avatar support until post-release
* Removed chatto and chatto additions pods
* Removed segue to conversations table view controller
* Removed choose avatar button in registration 2 window
* Removed avatar selection in settings view controller
* Removed avatar property from User class
* Converted switch shoutout code to swift 4.2

9/11/18

* Added functionality to delete friend requests
  + Allows the requestor to re-submit the request
* Added an observer in HomeViewController for incoming friend requests
  + Adds a badge to the Friends tab bar item; disappears when user navigates to friend request table view controller

9/14/18

* Added a “read” key/value to the friend request packet
  + The query for new friend requests will only fire on friend requests that haven’t been read, which resolves the issue of existing friend requests being re-observed at app launch
* Normalized data for beacons to be more easily accessible for observers/notifications
  + Added beacon table that contains UIDs and holds values of [friendUID:game\_title]
  + This data is removed from friends’ beacon table entries when the remove beacon button is pressed

9/18/18

* Fixed a bug causing an SVProgressHUD loop upon creating a new user with no games, friends, or beacons
* Fixed a bug that caused friend beacons not to appear after signing in until the friends view was loaded

9/22/18

* Cleaned up code and added observer for beacons
  + Bug: when the child node gets marked as “read”, the .childRemoved observer gets triggered

9/25/18

* Moved the beacon removal observer creation to its own function that is called on viewWillAppear AFTER the app checks for beacons to be marked read
  + Added a line to remove the removal observer BEFORE this check to avoid creating multiple observers for the same purpose
* Added a check to make sure that the removal observer wasn’t nil when attempting to remove
* Added ‘Pull to Refresh’ functionality to FriendsViewController
* ~~Fetched user data for SettingsViewController in HomeViewController initial load to avoid fetching data from firebase and seeing SVProgressHUD~~

9/26/18

* Moved code to save email to user defaults to RegisterViewController
* Moved code to save friend code to user defaults to Register2ViewController
* Moved code to save nickname to user defaults to Register2ViewController
* Fixed crash that occurred after reloading games from SettingsViewController
  + Issue was caused by attempts to access UI elements on HomeViewController that weren’t loaded
* Cleaned up viewDidLoad() in HomeViewController
  + Created function setUIElements() to house initial UI adjustments and settings
* Created function fetchUserInfo() in HomeViewController to check if user data has been loaded into user defaults. If not, it downloads the data from firebase
* Fixed the bug causing friend beacons not to populate when logging out/logging back in or logging in for the first time
* Removed the firebase call from pressing Create Beacon to reduce network usage

9/27/18

* Created RedownloadProtocol for use by SettingsTableViewController to make a call to redownload games without creating an instance of HomeViewCntroller()
* Reworked adding friends to not pre-load all users and use a query instead
* Removed beacon from user defaults when remove button is pressed
* Restructured checkForGameListChanges, updateGameList, and pullGameDictionary
* Updated registration text to remove references to avatars
* Added code to refresh friends list when empty and showing DZNEmptyDataSet
* Added functionality to add a game to AllGamesList in user defaults if a friend creates a beacon for a game that the user has not downloaded yet from firebase

9/28/18

* Fixed a bug where the currentGameCount wasn’t updating which prevented some functionality from running when a game is removed from firebase
  + Took it out of user defaults and just checked game could from AllGamesList instead
* Fixed a bug where the beacon would display “None” if a game was removed from firebase
  + It resets the UI to have no beacon but it does so after loading so it is noticeable by the user
    - Counting as a non-issue given that games will not be removed from the app, but will note it in the TODO document
* Rewrote game image sourcing to pull from local assets rather than from firebase
  + Removed images from firebase storage
* Fixed bug where one beacon removal observer would be created for each friend beacon that existed
* Reworked beacon observer to append the new game and reload the tableview instead of calling checkForFriendBeacons() again to avoid some unnecessary and troublesome re-running of code

9/30/18

* Installed SkeletonView pod
* Updated pods
* Implemented skeletonview; not an ideal solution. Going to go back to svprogresshud

10/1/18

* Removed view.tag if blocks from HomeViewController as they are no longer needed with SettingsViewController using a delegate method to redownload games
* Moved friend beacon loading to after new games are downloaded
  + Pushes back loading beacons, but beacons will now load with images at the same time
* Implemented search bar in AddGamesTableViewController
* Added Alamofire and AlamofireImage pods
* Added URL support for games added to firebase to pull game images from nintendo’s site
* Added background image to HomeViewController to show current beacon game image

10/2/18

* Re-implemented SVProgressHUD in HomeViewController initial load to prevent the user from creating a beacon before MyGamesList is populated
* Added a contional for DZNEmptyDataSet when searching through all games to display an alternate message
* Fixed issue where beacon label in HomeViewController was cutting off large words in the middle of the word
  + Solution taken from <https://stackoverflow.com/questions/23121240/ios-uilabel-autoshrink-so-word-doesnt-truncate-to-two-lines/47469960#47469960>

10/3/18

* Added next/done return key functionality to LoginViewController
* Added code to remove observers when user logs out
* Added next/done return key functionality to both registration view controllers
  + Added custom next button to friend code text field in the second registration view controller due to the keyboard being a numberpad

10/4/18

* Purchased Apple Developer Program membership
* Integrated app with onesignal
  + Added pod, code in appdelegate.swift
* Created various certs
  + Put them in Switch Shoutout/Documents/certs
  + Backed them up to dropbox
* Moved push notifications prompt to HomeViewController.viewDidLoad()
* Changed uid property of User class to fbuid
  + Changed all references
* Added player\_id property to User class
* Added OneSignal subscription observer to homeViewController to look for subscription changes
* Signing out now removes the current user’s beacon from firebase (along with instances of it in friends’ beacon lists)
* Friend requests now send player\_id
* Added functionality to send out notifications for beacons

10/5/18

* Added push notifications to friend requests
  + Added increased app badge count for friend requests only
* Changed the OneSignal subscription observer to only fire code when subscription changed to true
* Added a Notification Center observer to AppDelegate.swift
* Moved the firebase beacon (adding) observer code to onDidReceiveData
* Silent notification is now sent out when a user removes a beacon
  + Moved remaining beacon removal observer functionality to onDidReceiveData
* Utilized Beacon class/object throughout code to simplify things
* Moved OneSignal subscriptionChanged observer to AppDelegate.swift and made it create a Notification Center post to run the code to update the player\_id in firebase that resides in HomeViewController
* Made subscriptionChanged observer update the player\_id in user defaults
  + Sets new bool “player\_idWasUpdated” in user defaults that HomeViewController checks right before the notification observer is created
  + Basic operation flow:
    - If player\_id is updated BEFORE HomeViewController notification center observer is created:
      * HomeViewController calls new function checkForUpdatedPlayer\_id() which updates player\_id in firebase
    - If player\_id is updated AFTER HomeViewController notification center observer is created:
      * AppDelegate.swift posts a notification center notification which is observed by HomeViewController and then calls checkForUpdatedPlayer\_id

10/8/18

* Moved friend request observer functionality to the Notification Center observer onDidReceiveData
  + Only creates the badge icon on the friends tab
* Changed OneSignal.inFocusDisplayType to OSNotificationDisplayType.none to not display any alerts when the app is open
* Added functionality to automatically navigate to the friend request view upon tapping a friend request notification

10/9/18

* Fixed bug where parameters were passed out of order when creating a beacon to display when checking for friend beacons
* Fixed bug where player\_id update notification center notification was being called after the user logged out because the player\_id was being removed from user defaults
  + This caused the app to think that a new unique player\_id had been issued since there was no existing player\_id to match it to
* Cleared up confusion about a pseudo-bug where the current beacon was not loading correctly upon closing and reopening the app
  + Per <https://stackoverflow.com/questions/2622754/why-is-nsuserdefaults-not-saving-my-values>:
    - “If you terminate your app by pressing the home button (in the Simulator or on the device), your NSUserDefaults will get saved.  
      If you terminate your app by pressing "Stop" in Xcode (in the Simulator or on the device), your NSUserDefaults *might* get saved, but there's a good chance they won't.”
* Created archive of app and uploaded to iTunes connect
  + Need to keep build and version numbers consistent between both app targets
* Created privacy policy through iubenda.com
* Removed the home tab item badge restriction when viewing the home screen
* Added beacon id to firebase when creating beacon
  + Fixed bug where friend beacons created when user had app closed would not be removed when user opened app and friend removed beacon
* Archived build 1.0.1

10/29/18

* Added fitToAvoidWordWrapping to game cellLabels in MyGamesTableViewController and AddGamesTableViewController to fix text getting cut off
* Added a status message “Updating Game List…” to SVProgressHUD when all games list is being updated
* Added functionality to remove beacons from each other’s’ firebase beacon lists when a friend is deleted
  + Doesn’t update UI – should revert to firebase observers

2/23/19 (The great return)

* Made friendstableviewcontroller reload table on viewWillAppear()
  + This automatically updates the friend list after confirming a friend request and navigating back to the friend list