Group 5
3/6/2023

Instructions and Notes for Running

- We have two example accounts that are fully set up to use the app
    a. Username: Player 1, password: pass1
    b. Username: Player 2, password: pass2
- To test the non-battle screens of our app, you don't need to run the server. Just click the sign in button and log in as one of the two accounts listed above.
- To test the battle screens, first run the server and then log in on our app. Swipe left from the home screen twice. You can click either "join team" or "create team". In beta these will lead to different screens but for now both take you to the battle screens. The game is hardcoded to need two players right now to move past the initiative screens.
- To run the server, just open the project on xcode and press run. For now, you need to restart the server between every run of the game but this will be fixed in beta.
- While in battle, a lot of the spells don't work yet (we're implementing them in beta) and clicking on them might cause the game to crash. The "fight" option will definitely work.

Contributions:

Jalyn (25%):

- Built the character classes and subclasses for different character classes
- Builts Weapon and Armor protocols as well as all the structs that extend these protocols for equipping and wielding
    - Implemented functions that allow user to wield/wear armor no matter there proficiency. If the user's class is not proficient with a certain weapon or suited for a certain piece of armor their damage or defense is weakened respectively.
- Co-designed the flow of taking turns with Kelly, including what attributes need to be sent and retrieved from the server at what point in the flow.
- Built Item protocol with all the items that extend this protocol
    - Implemented functions to permit user to use each item on themself or on an ally/opponent
- Built 100 tests to test the Character classes and subclasses, weapon and armor functions, item usage, and message logging. Some no longer work due to a project restructuring, but shall be reconfigured for beta release.
- Built MessageLog class which will display the actions each user takes during battle to be displayed to the idle users while they are waiting for their turn
- Implemented the functions respective to each action a certain character class can take. Then, connected them with the front end so that clicking on a certain action will do the following effects on the person who took the action and (possibly) their target.

Kelly (25%):

- Designed flow of combat and team matching with Jalyn
- Set up database on firebase and integrated it into our app
- Created a server to coordinate player order and taking turns
- Implemented the ability to take turns for combat
- Implemented backend for login, registration, initiative, and display initiative screens
- Implemented the firebase part of the backend for messages seen while on the idle screen
- Wrote code to update character information on firebase between turns and read updated information to store locally

Nick (25%):

- Helped create the design of each screen and contributed to the project's wireframe
    - Modified the wireframe to maximize readability
- Created extension functions for the class UIViewController to improve the process of creating UI components
    - Creating buttons, labels, image views, etc.
- Created the Login and Registration screens
- Created the Class Selection screens
- Created the Stats menu screen
    - Pulled data from our global Character object to populate the display
- Created all three Inventory menu screens
    - Converted our Weapon, Armor, and Item (consumable) objects into inventory structs that are displayed as cells on the screen's TableView
    - Implemented a segmented view that repopulates the TableView with different data (depending on the segment selected)
- Created the Battle menu screen
    - Added a swipe gesture to the three menu screens above
- Created the Roll to Hit battle screen
- Created the Notification banner

Alekhya (25%):

- Helped create the design of each screen and contributed to the project's wireframe
- Created all the table views (4) and the scroll view for the battle screens
    - Action table, Item table, Equip Table, and Stats Display Table
    - Created custom table view cells for all of the table views
- Created 4 of the Battle Screens
    - Action, item, equip, and idle
- Created the Party Menu Screens
- Created the Code Entry Screens
- Created Team Matching screens
- Created Roll Initiative Screens
    - Used Sprite Kit to create the dice rolling animation for the roll initiative screen
- Created Battle Results Screens (Victory and Loss)

- Created all of the design for the character sprites using a sprite building website
- Found or created most of the icons for items including weapons and armor
- Added the textfield delegates to remove the software keyboard

Differences:

- Pushed out keeping track of user steps
    - Why?: We have not reached the portion of class and we didn't have time to self-teach ourselves how to pull this information from HealthKit.
- Have animation for 2 classes complete:
    - Why?: We decided to create the screens programmatically which slowed down the creation process and creating the animations was not a priority to creating all the necessary screens.
        - We completed the dice animation but not the character animation
- Character creation
    - Why?: We decided to create the screens programmatically, which slowed down the creation process. This delayed the process of adding the functionality associated with creating a character
- Notifications
    - Why?: We decided to create the screens programmatically, which slowed down the creation process. This delayed the process of adding the functionality of item drops, which notifies users
- Roll Dice and Roll Checks
    - Why?: We completed this early because we realized how closely it related to completing actions during combat.