

# WARGAME PROGRESS REPORT

Week 8: March 27th - April 2nd

*Reid Coates* | Client Coordination and Backend Development Lead

*Jack Kelley* | Organization Lead and Frontend Development

*Alexander Hassan* | Testing Lead and Frontend Development

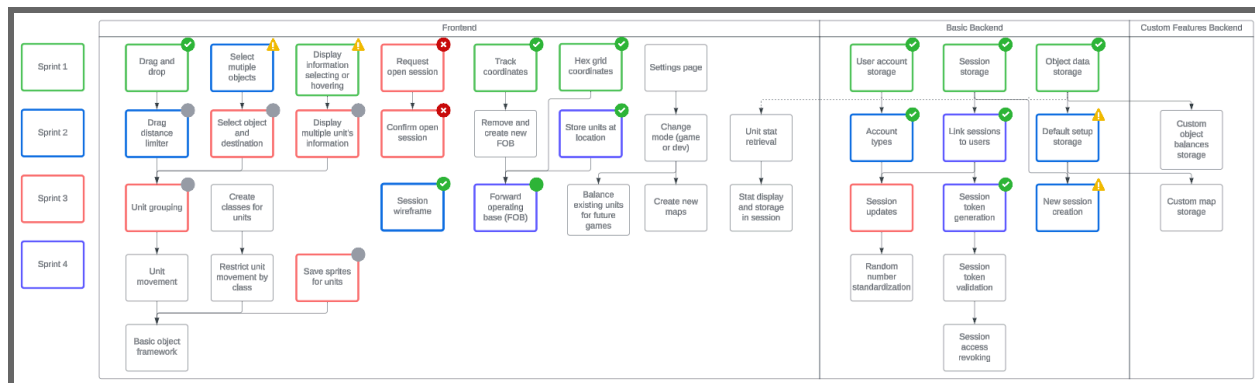
*Luke Muilenburg* | Frontend Development Lead

Group 23 | sddec24-23 | EE/CPRE/SE 491

Client: Reid Coates (AFROTC and Major Stephanie Jones)

Advisor: Ahmed Shakil

# WEEKLY PROGRESSION



## Frontend to Backend HTTP Requests - **Jack Kelley**

- With the progress that was made on the backend, we are now able to store some basic game values such as sprite grid locations. On the frontend this week, we worked on making requests to our server that update grid locations in our database based on their location on the frontend grid, as well as updating the location on the frontend based on the locations stored in the database. This will be used for sending moves to an opponent and receiving an opponent's moves once turns are implemented.
- The framework for making the requests is in place and the framework for updating sprite locations based on retrieved locations from the database is also in place. This coming week, we will be connecting those 2 parts to be able to submit moves and update grids based on the retrieved location.

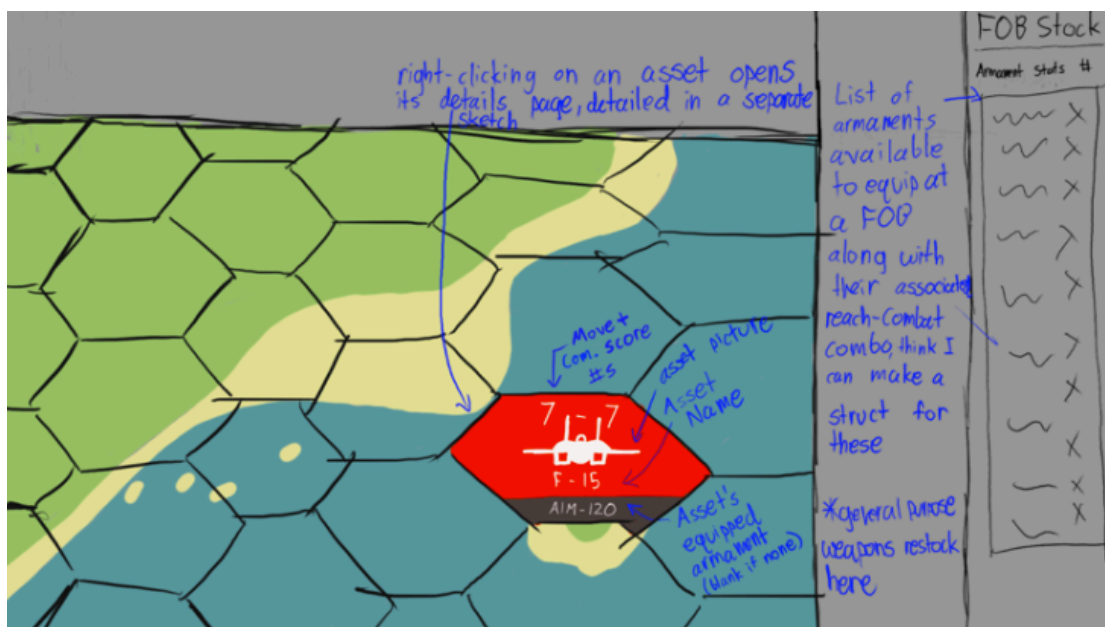
## Frontend to Backend HTTP Login Request and game start menu - **Alexander Hassan**

- Created a room with 2 different animated buttons: Quit, which exits the game, and Login, which takes a username and password set of strings from the respective input textboxes that hover above the login button. The login button sends a POST HTTP request to the backend server with a JSON string containing the user's typed-in username and password.

- the input text boxes are their own objects, which limits the user's input length to 30 characters (this can be changed by adjusting one variable), and include a blinking cursor for UI purposes.

### Asset Management System (Frontend) - **Luke Muilenburg**

- Created screen sketches to scope the work that needs to be done to get the asset management system working. These are attached below.
- Started work on the frontend code for the system.



### Web Handlers - **Reid Coates**

- Added more HTTP request handlers to the account and game controllers in the MVC model.

### Account and Join Code Generation - **Reid Coates**

- Accounts are almost able to be created via a frontend request. The framework is designed in the database but the handlers are not implemented. Join codes for new games can now be created via random six digit codes which are cross referenced against each other to ensure uniqueness and removed when used.

### Database Finalization - **Reid Coates**

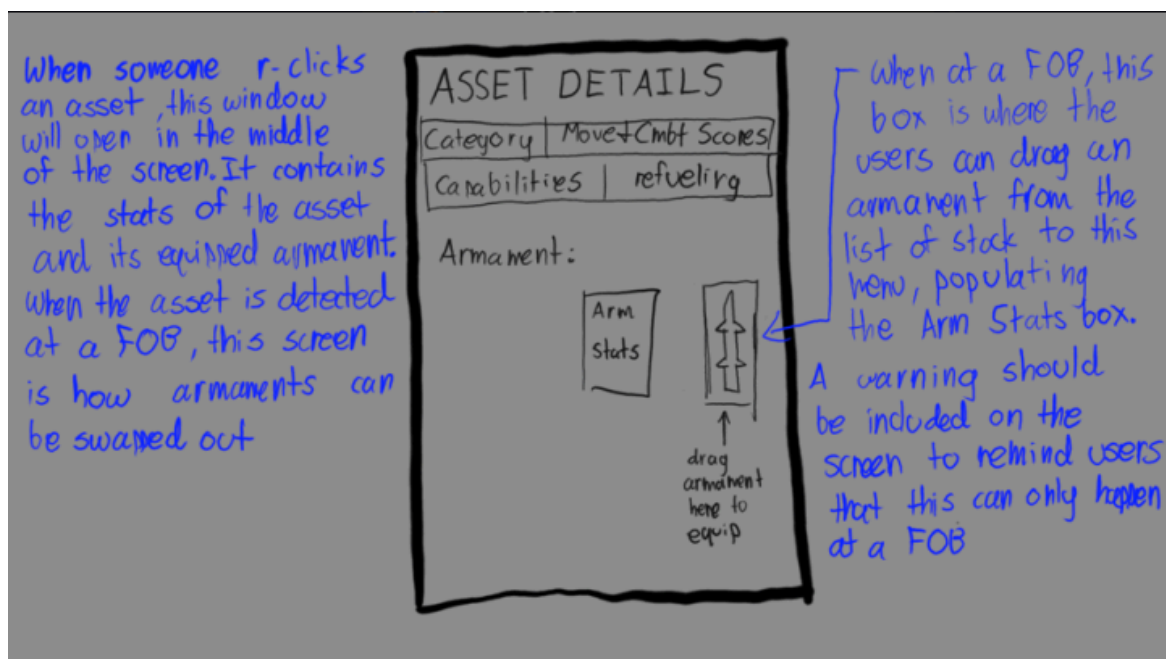
- Fully implemented MariaDB and setup TCP connections to Java app deployed on server and locally which has allowed development to use the real database for testing and forward movement. Schema rules and templates have been tested and are available for extension in the deployed Java app.

### Firewall Rule Creation - **Reid Coates**

- Added ports 3306 (DB/TCP) and 8080 (HTTP/TCP) to firewall on Linux server as permanent connections allowing outside access to database from account login via Heidi SQL or a similar app. Accounts created to give the team developer access to database.

### Postman Request Rework - **Reid Coates**

- Postman requests are now organized in a fashion which allows all team members to clearly see which requests are still working and which are being serviced.



## PENDING ISSUES

### Gameboard Scaling

- Our original intent was to provide the ability to scale the size of the game board grid to better fit whatever map the user may choose to upload, but the capabilities of GameMaker Studio have made developing a hexagonal grid more complicated than we originally thought. We need to decide how we want to go about implementing the scaling or if we want to scrap the idea altogether.
- Note: This has been here for multiple weeks at this point because it is an important design choice that needs to be made eventually, but isn't an issue that we can confidently resolve at this time.

### Should we implement sending HTTPS requests as opposed to HTTP requests?

- In the final product delivery, it may become necessary to encrypt all network communications with HTTPS requests. However, this may require significant rework and may not entirely be possible with Game Maker.

# INDIVIDUAL CONTRIBUTIONS

Team Member	Contribution	Weekly Hours	Total Hours
Alexander Hassan	Worked on sending HTTP requests to the backend and created a login screen with multiple menu options.	10	42
Jack Kelley	Worked on frontend connection to backend. Implemented framework for updating sprites based on an input grid location, worked on framework for making HTTP requests in GameMaker, and updated hexagonal grid logic.	10	46
Reid Coates	Created more http handlers, finished database setup, created account and join code schema, random, unique join code creation on demand,	15	61
Luke Muilenburg	Generated screen sketches and began work on the frontend for the asset management system.	4	34

# NEXT WEEK

Task	Members	Completion Date
Finish Implementing the user's ability to select multiple assets, highlight updates, and drag distance-limiting for the user, according to the rulebook.	Alex	4/9
Implement a menu on the frontend for the user to be able to create an account and send typed-in information to the backend system to be stored in an accounts table in the database.	Alex	4/9
Bug fix asynchronous web traffic proof of concept.	Reid	4/9
Setup account creation and log in handlers based on the existing account db framework.	Reid	4/9
Create game logic and coordinate math framework	Reid	4/9
Complete the FOB stock segment of the Asset Management System and make a struct to store asset equipment	Luke	4/9
Make connection between frontend and backend for sending and retrieving sprite coordinates.	Jack	4/9
Begin implementing game-turn logic for frontend	Jack	4/9