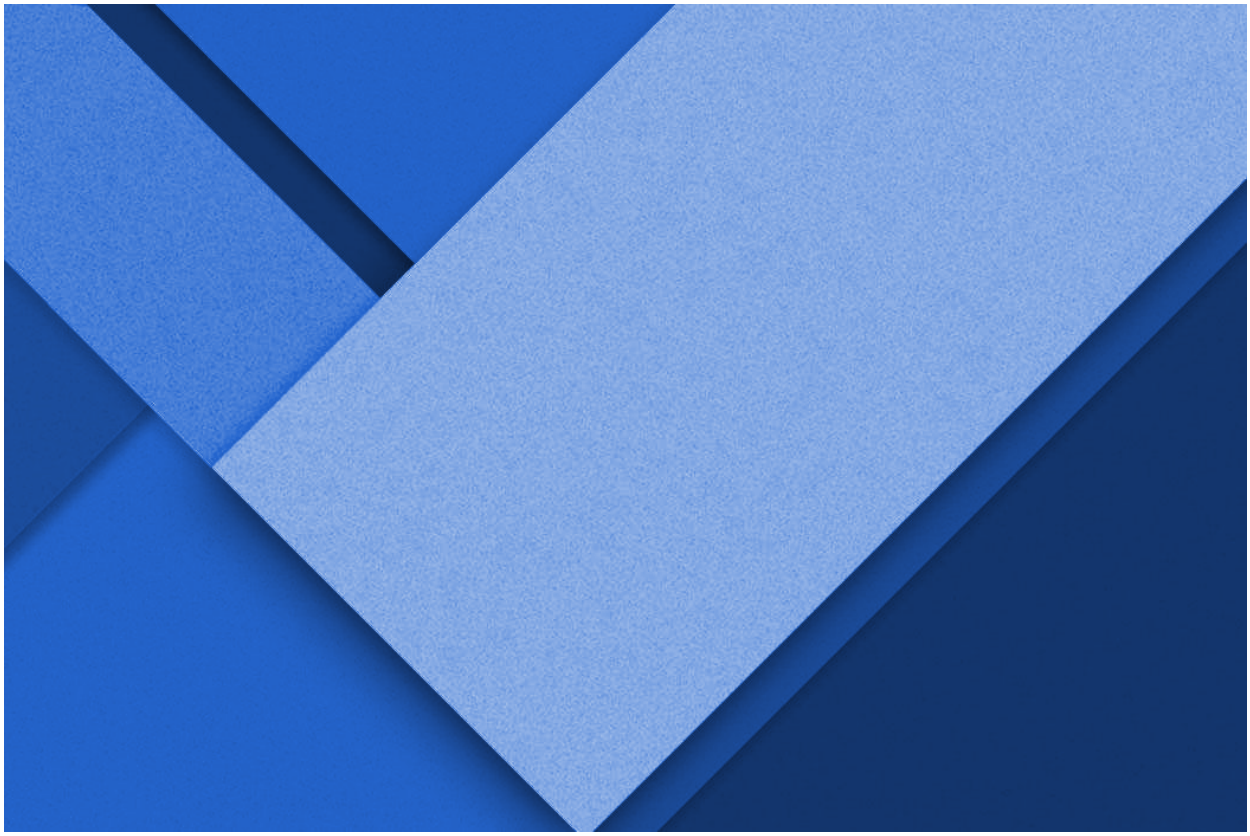


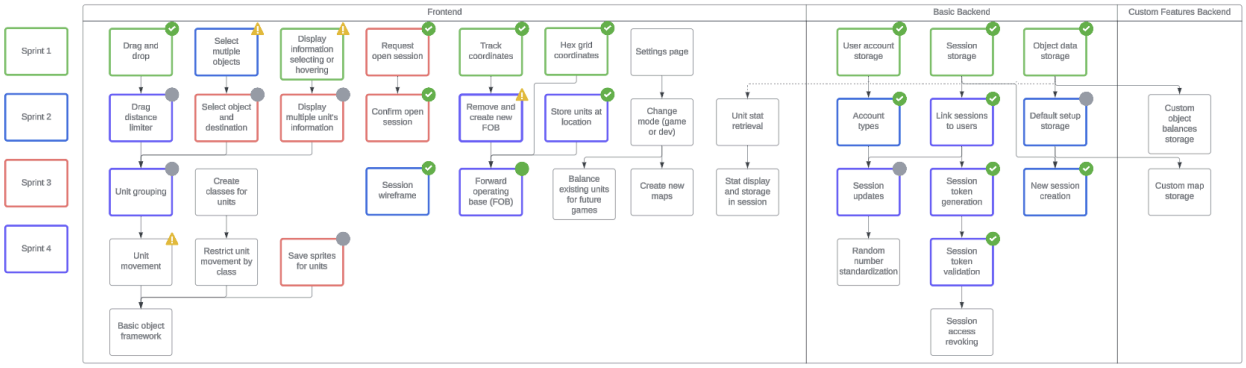
WARGAME PROGRESS REPORT

Report 2: September 5th - September 19th

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 492
Client: Reid Coates
Advisor: Ahmed Shakil

WEEKLY PROGRESSION



Updated Movement to Point and Click - **Jack Kelley**

- We had originally implemented a drag-and-drop approach for moving assets, where a user would grab a tile on the game board and drag it around the board to the desired location for it. With the implementation of pop-up windows for displaying all assets on a tile, we decided it would make more sense for the user to select a tile to move, and then click on the desired new tile location. The usage feels very similar but makes more sense with our design direction.

Added Pop-up Windows for Tile Information - **Jack Kelley**

- To manage displaying all of the assets that are in the same tile, we decided to implement a pop-up window that appears when the user right-clicks on a tile that contains assets belonging to their country. The user is shown a list of all assets on that tile, including the asset's icon, name, stats, and buttons with options for game decisions.

SQL scripts for Asset stats and tables - **Alex Hassan**

- Wrote sql scripts to generate the rough draft of tables required for the backend to get asset data from the database and send to the front-end, Reid

took this rough drafts and re-created them using Java to initialize at the start of the server.

Helped Luke's Armament Management work - **Alex Hassan**

- updated the update_game HTTP request on the frontend to get the current game data being sent, and adjusted the asset data structure on the frontend accordingly.

Individual Aircraft Equipment System**- Luke Muilenburg**

- I worked on a gui and functionality for equipping individual aircraft. I styled it to look like an aircraft MFD. I also implemented a storage tab that shows available weapons that changes based on which team a user is on. Alex helped me greatly with starting to integrate http requests into this framework.

Update web request objects **- Reid Coates**

- New default assets and expanded information in regular assets are returned in web requests. Default lists of armaments are now pushed when a request is made.

Fix CI/CD pipeline = **Reid Coates**

- Add testing framework to CI/CD pipeline and adjusted deployment phase to only run if a backend developer is making the push and the push is stemming from the production branch.

Default game asset rework **- Reid Coates**

- Built in generation of default games including assets and a rework of the data sent to frontend to allow for a singular point of save data. This allows for the balancing of the game in the future at one place and the rapid deployment of

the database module. The default is hard-coded in one place to allow one repository of default assets in one location.

PENDING ISSUES

Gameboard Scaling

- After implementing game assets and assigning values to those assets (combat power, movement speed, etc.), we've found that some assets can move across the entire board very quickly. With this, we are considered rescaling the game more to include more tiles. We found, however, that this could take a lot more time than originally anticipated so we are now looking into other alternatives such as modifying the values for the assets, or applying a conversion from the base asset values to a value that is more suitable to our game board.

Turn Interfacing

- We need to add interfaces for guiding the user through playing a turn and the possible options they have on any given turn. More generally, we want to start with notifying the user if their opponent has submitted a move and is waiting on them to make a move, as well as display what turn it is. We also want to include more information about game assets and clean up the code for displaying where an asset can be moved to (its range).

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.
- Note: This change should not affect how the game communicates with the backend and will be a consideration that we will discuss and decide on finalization of the backend logic and frontend graphics.

INDIVIDUAL CONTRIBUTIONS

Team Member	Contribution	Weekly Hours	Total Hours
Alexander Hassan	Wrote sql scripts to generate the rough draft of tables for assets and armaments, and updated the update_game HTTP request on the frontend to get the current game data being sent.	5	72
Jack Kelley	Point and click movement implementation, pop-up window implementation	12	73
Reid Coates	Updated web request objects, added default asset and armament handling, updated and repaired CI/CD pipeline.	20	93
Luke Muilenburg	Work on MFD interface and red/blue team storage tabs	8	60

NEXT WEEK

Task	Members	Completion Date
Continue addition of default assets and armaments	Reid	9/26
Game logic for first dogfight between J10 and F16 on the frontend.	Reid	9/26
Update Submit move request to include the araments being equipped to assets and the available list of armaments with reduced quantity once equipped.	Alex	9/26
Help Luke with equipping Assets by manipulating front end data structures.	Alex	9/26
Sprite completion	Luke	9/26
Refine the equipment system	Luke	9/26
Merge all front-end code to reflect movement changes	Jack	9/26
Implement FOB for armament prompts	Jack	9/26