

WARGAME PROGRESS REPORT

Report 1: August 29th - September 5th

Reid Coates | Client Coordination and Backend Development Lead

Jack Kelley | Organization Lead and Frontend Development

Alexander Hassan | Testing Lead and Frontend Development

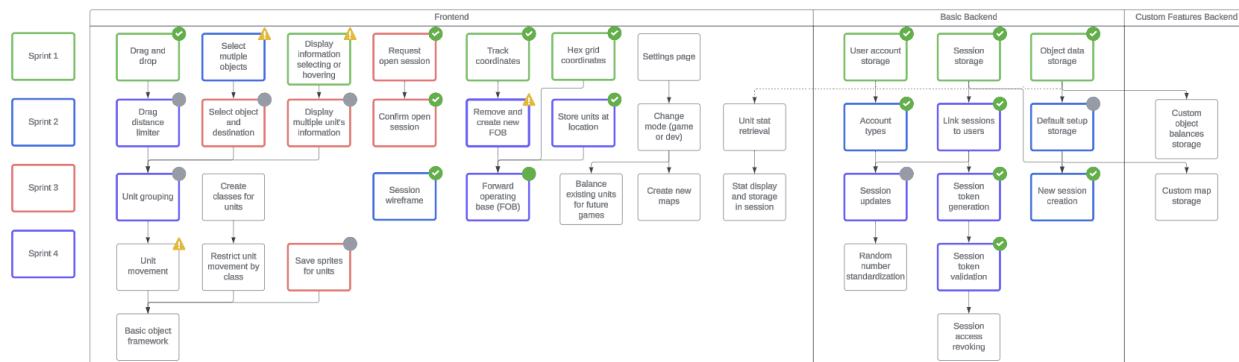
Luke Muilenburg | Frontend Development Lead

Group 23 | sddec24-23 | EE/CPRE/SE 492

Client: Reid Coates

Advisor: Ahmed Shakil

WEEKLY PROGRESSION



First Team Meeting of the Semester - **All Members**

We conducted an all team meeting on September 2nd to get up to speed on our plans for the next 2 weeks. We made progress on our respective tasks this week and plan to supply complete deliverables at the end of next week. We are also coming to grips with the new course structure to ensure the success of our project.

Codebase Refamiliarization and Troubleshooting - **Jack Kelley**

- Most of my first week has been spent getting refamiliarized with the code base and GameMaker studio. We also organized Gitlab and set up a protected main branch that we will be merging off of for future work.
- While testing our main branch, we encountered some internal errors with the api and spent some time trying to figure out what was wrong to get a functional main branch in place for us to be able to branch off of confidently moving forward.

Asset Sprite and stats implementation- **Alex Hassan**

- Took physical wargame assets from the rulebook and pdfs from our cybox drive and included that information into our current asset struct. After discussion, this data should be stored on the backend database, as well as the enumerations that make the newly added code meaningful.

Generate Sprites for Game Pieces- **Luke Muilenburg**

- Now that we have completed some of our round trips at the end of last semester, we have begun to work on the aesthetic portion of the project. The best place to start on this is to make the sprites for the moveable game pieces.

Continue Work on Equipment System- **Luke Muilenburg**

- I was able to do some work over the summer on the equipment system, but I did a poor job of integrating it with code from other team members. Over this week and next I am working to add functionality and get it integrated into our git environment.

Join code, game adding, and player country allocation - **Reid Coates**

- When a new game is created, a joinCode is generated (unique 8 digit code) and the game is added to the creating user's account. When the join code is used by a different player, the game is added to that user's account as well. The turn flag is then set to allow either player to update a move in the database and start the game logic loop. The joinCode is then removed from the game file and replaced with a -1 signaling the file to be unjoinable since it already contains both player one and two.

Delete game from all - **Reid Coates**

- This feature allows a user that owns a game to delete the game. Since this is a two player game, removing a player in an already created game renders the game unplayable. It is deleted from both user's accounts when either user deletes it. The game is also removed from all database tables associated.

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 considering 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	Added frontend code to store stats of assets and armaments; in the process of transferring this data to the backend to persist in our database.	10	67
Jack Kelley	Code base familiarization and submit move internal error troubleshooting	8	61
Reid Coates	Creating default game creation. Created a join game feature and added new games to accounts as well as auto assigning players on join.	7	73
Luke Muilenburg	Work on sprites for game pieces and familiarization with the code for the equipment management system.	4	52

NEXT WEEK

Task	Members	Completion Date
Add all Asset and Armaments information in the Database, and add requests/controllers to the front/back end systems respectively to make this data accessible.	Alex/Reid	9/12
Default game creation database initialization and drop/re-add feature for app startup.	Reid	9/12
Game logic for first dogfight between J10 and F16 on the frontend.	Reid	9/12
Sprite creation	Luke	9/12
Refine the equipment system	Luke	9/12
Implement "opponent waiting" flag	Jack	9/12
Implement turn counter	Jack	9/12