

P02: Makers Makin' It, Act I

Conquest by RightMouseButtonS

ROSTER: Sean Zheng (pm), Bogdan Sotnikov, Robert Chen, Mottaqi Abedin

TARGET SHIP DATE: 2026-01-16

Idea Summary

A simplified version of Risk involving 2-6 players. Players start out with a set number of armies and take turns placing their armies across 42 territories on a world map until all armies have been placed. At the start of each turn, each player places additional armies (territory / 3 rounded down but at least 3 armies) and can choose to attack and move troops across adjacent/connected territories. When attacking, the attacker chooses a territory to attack from and an adjacent enemy territory to attack and can continue until they only have 1 army left on that territory, when they choose to stop, or when the defender loses all of their armies in that occupied territory, after which the attacker takes over that territory. During each turn, players can earn bonus armies if they have continent bonuses, earned by owning all territories in respective continents. The game ends when there is one player standing.

Program Components

- HTML
 - base.html
 - navbar/header for pages
 - menu.html
 - Where user can choose how many players and read the rules
 - The starting page
 - Transitions to the game page with play/start button
 - game.html
 - Loads existing game or start new game
 - Main game interface
 - Handles user interactions
 - Interactable map with 42 separate territories
 - login.html
 - Allows users to log in
 - Checks login info from database
 - register.html
 - Allows new users to create accounts
 - Adds new login info to database
- DBs
 - territories.db (will contain info like connected territories and continent groups)
 - games.db (will contain the current state of the game)
 - updates after every turn

- will contain data such as how many armies each player has and how many armies are on each territory (and whose).
- users.db (user account info)
- Python Files
 - __init__.py
 - Runs flask
 - Calls helper functions from other python files and connects to js with jsonify
 - The one that runs the games during each turn
 - game.py
 - Contains helper functions used in __init__.py including:
 - setup
 - availableMoves (for reinforcing, attacking, and selecting)
 - addTerritory (for reinforcement)
 - attackTerritory
 - Creates and utilizes/updates db tables
- Static Folder
 - JS
 - game.js
 - load
 - attack
 - move
 - gameplay
 - helper functions

Database Organization

territories

| | | |
|---------|-----------|-------------|
| INTEGER | id | PK NOT NULL |
| TEXT | name | NOT NULL |
| TEXT | connected | NOT NULL |
| TEXT | group | NOT NULL |

games

| | | |
|---------|--------|-------------|
| INTEGER | id | PK NOT NULL |
| TEXT | armies | NOT NULL |

| | | |
|------|-------------|----------|
| TEXT | territories | NOT NULL |
|------|-------------|----------|

users

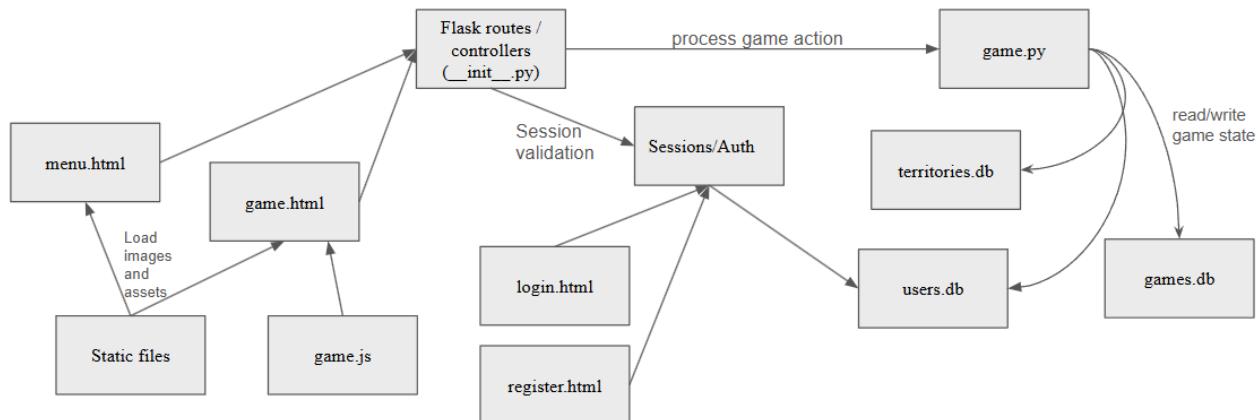
| | | |
|------|----------|-------------|
| TEXT | username | PK NOT NULL |
| TEXT | password | NOT NULL |

(other stats TBD)

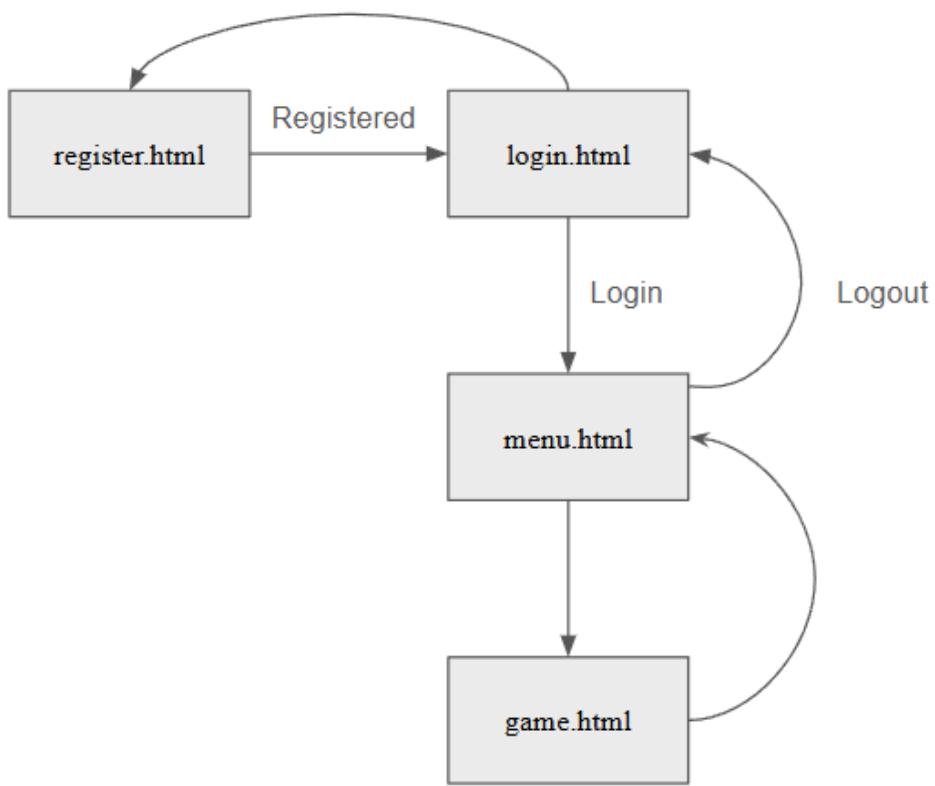
Chosen FEF: Bootstrap

We will be using Bootstrap because it comes with buttons and modals, which will be useful for making an interactive game page. It is easier to learn and utilize than Tailwind and foundations, meaning more time to focus on game logic and functionality.

Component Map



Site Map



Task Breakdown

| <u>Task</u> | Devs | Completion Status |
|--------------------|-------------|--------------------------|
| FEF | Robert | D |
| Flask and JS | Mottaqi | D |
| Database | Sean | D |
| Piece Movement | Sean | D |
| Attacking | Bogdan | D |
| Game setup | Bogdan | D |
| Log in | Mottaqi | D |