



A GAMING EXPERIENCE

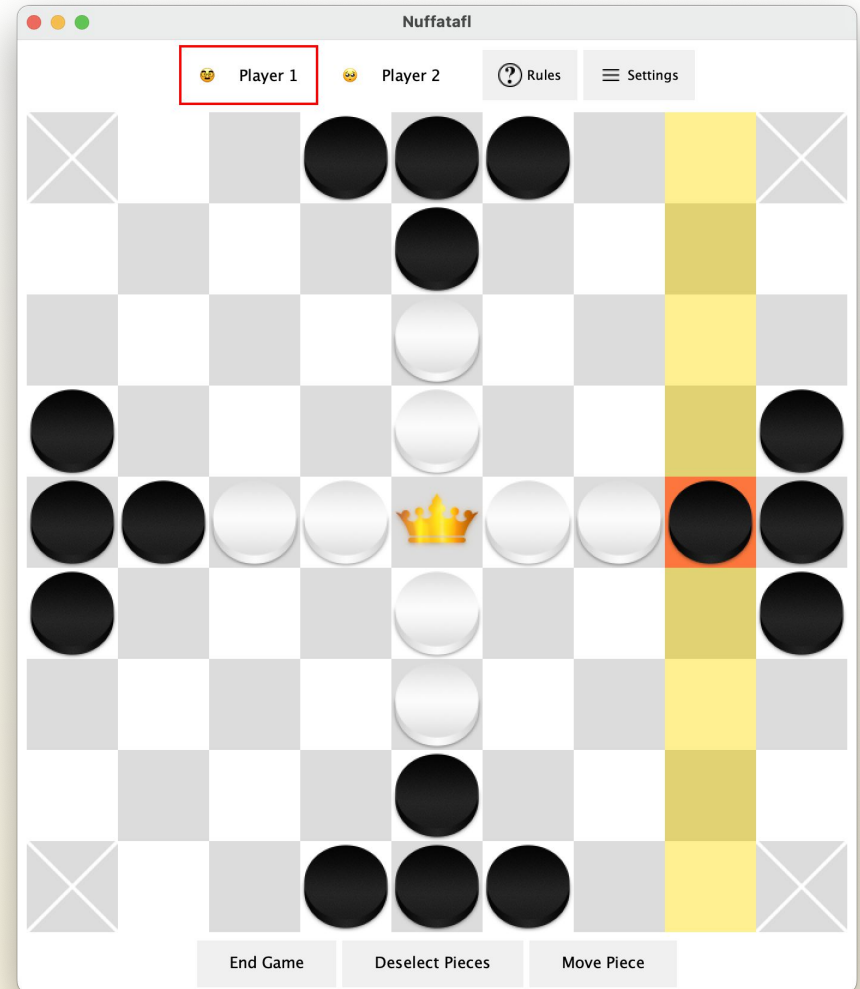
Presented by:
Mark of the Chili Onions
Mark, Cash, Orion

Project Overview

Nuffatafl is our version of Tablut, a chess-like game.

Features include full two player gameplay, theming, and different modes for display.

Our UI design was limited by Java Swing's capabilities and our input system is sufficient but there is room for improvement.



Game Description

- Tablut is played on a 9x9 grid
- 2 teams - Attacker and Defender
- Each piece moves like a rook in chess
- Defenders win when king reaches edge of board
- Attackers win when king is surrounded
- You can capture pieces by surrounding them in various ways



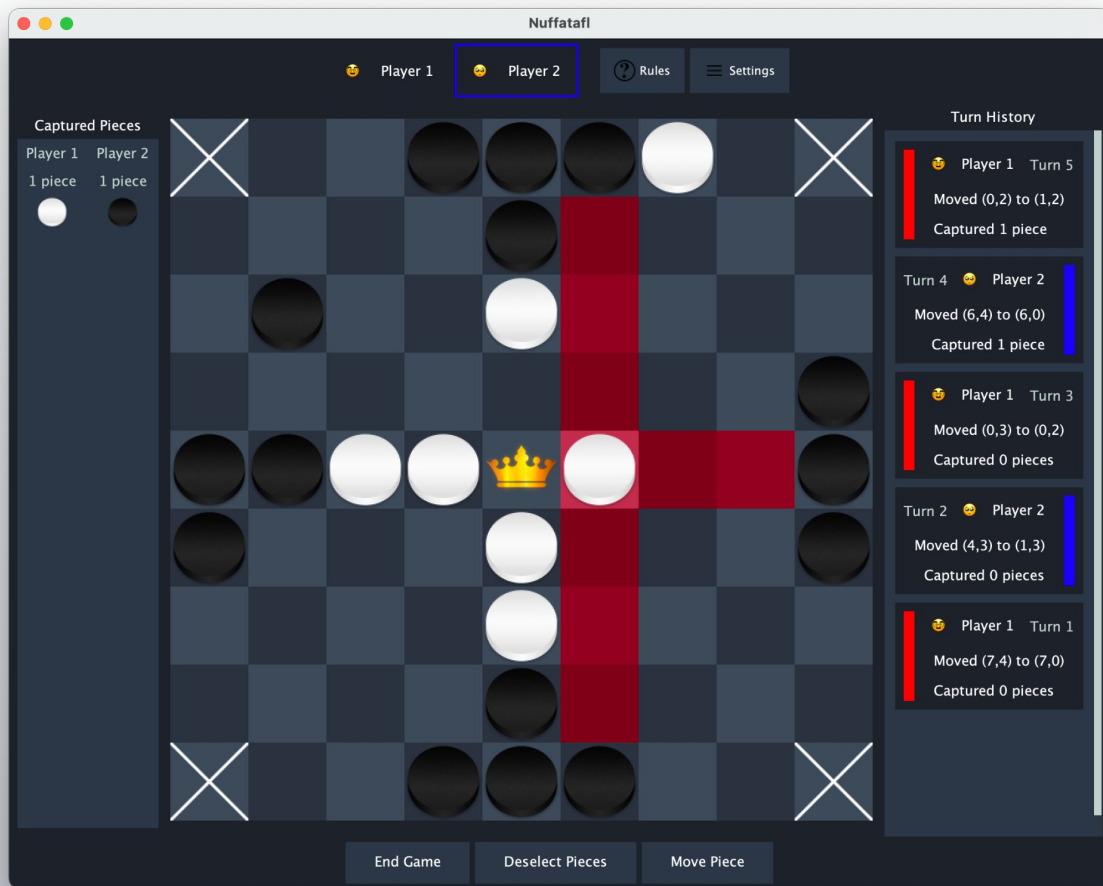
Project Requirements

- Each side should win according to the rules of the game
- Turn history should be tracked and displayed
- The user should be able to choose which theme they would like to use

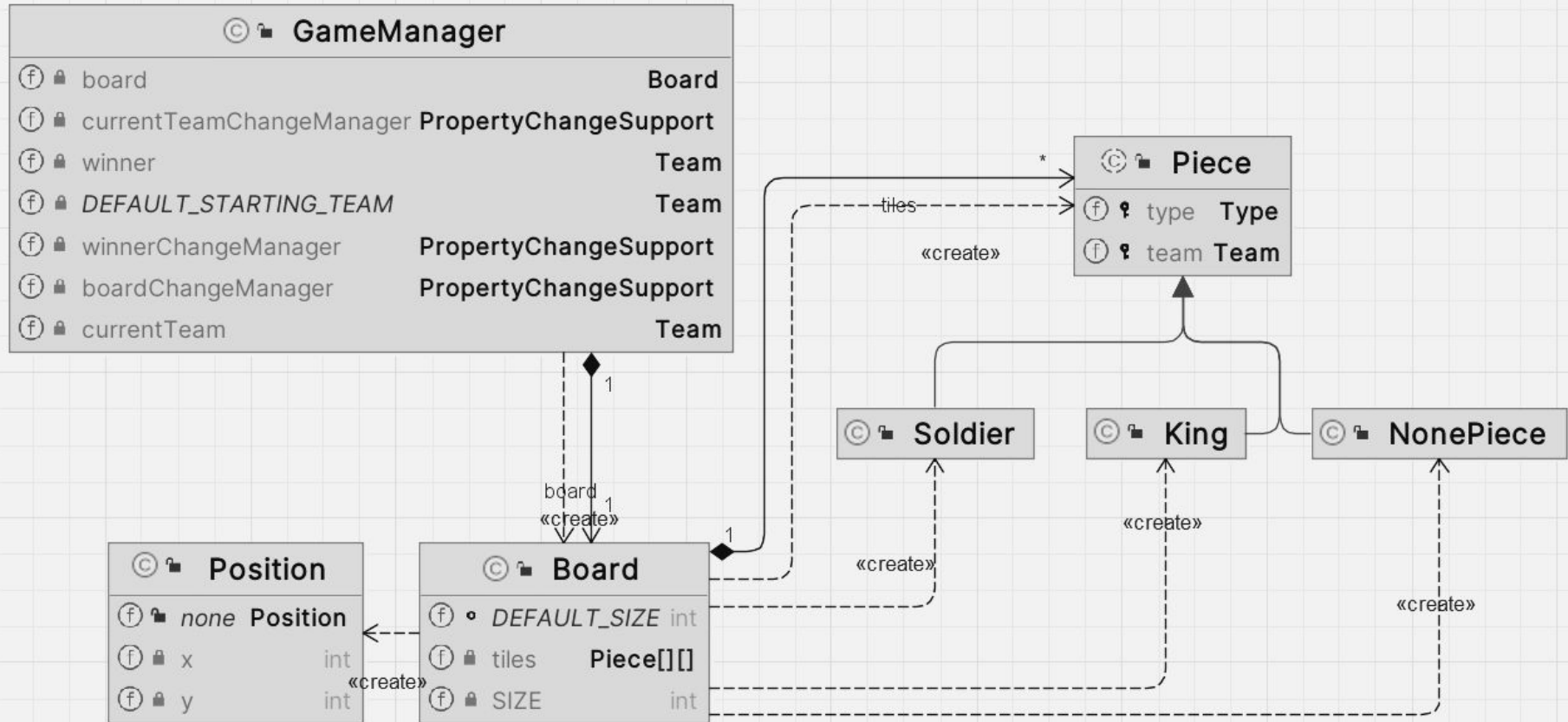


Project Solution Approach

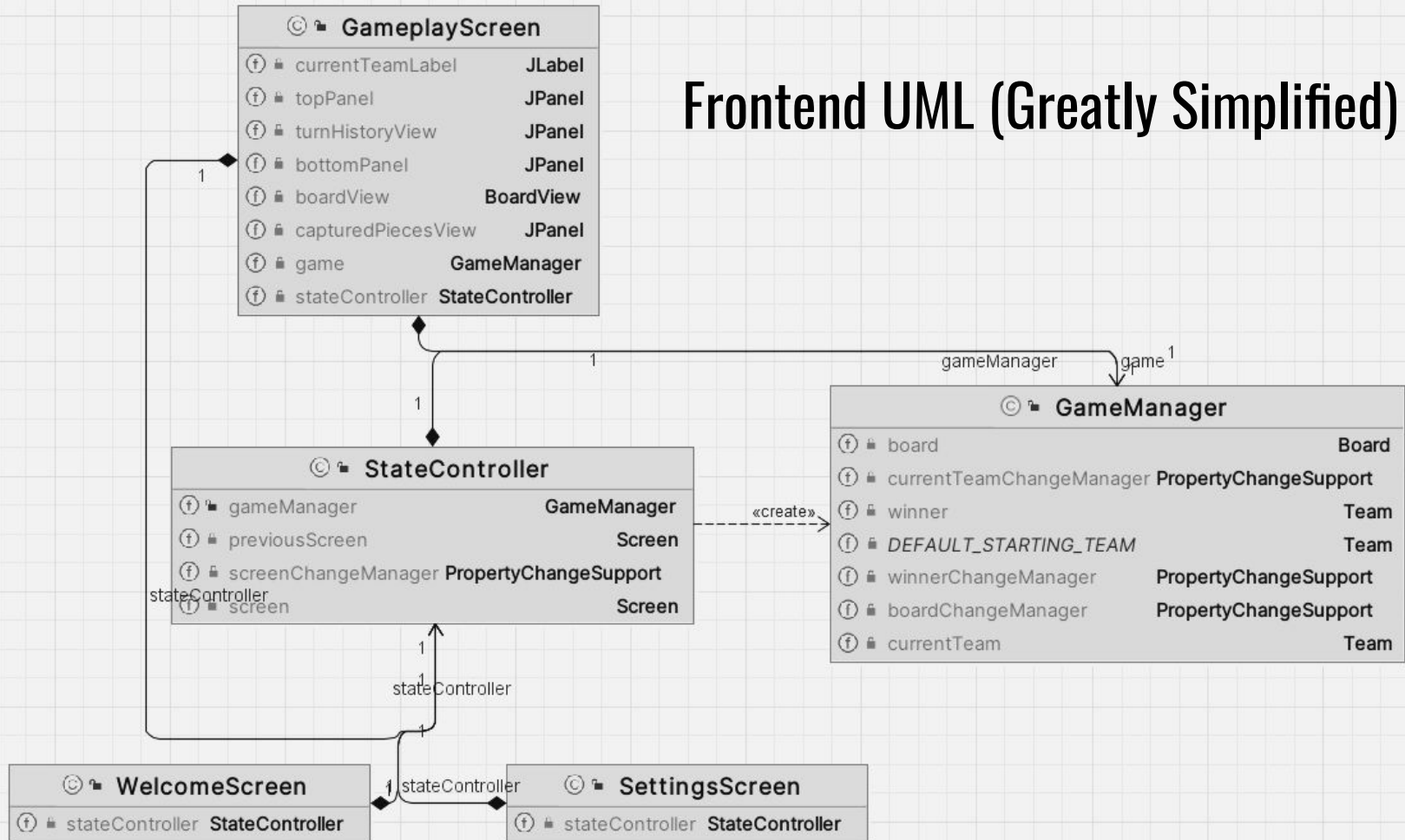
- MVC
 - Model: Board
 - View: MainView
 - Controller: StateController, GameManager
- Observer patterns for theming and updating board
- Proper rule implementation
- User experience



Backend UML (Greatly Simplified)



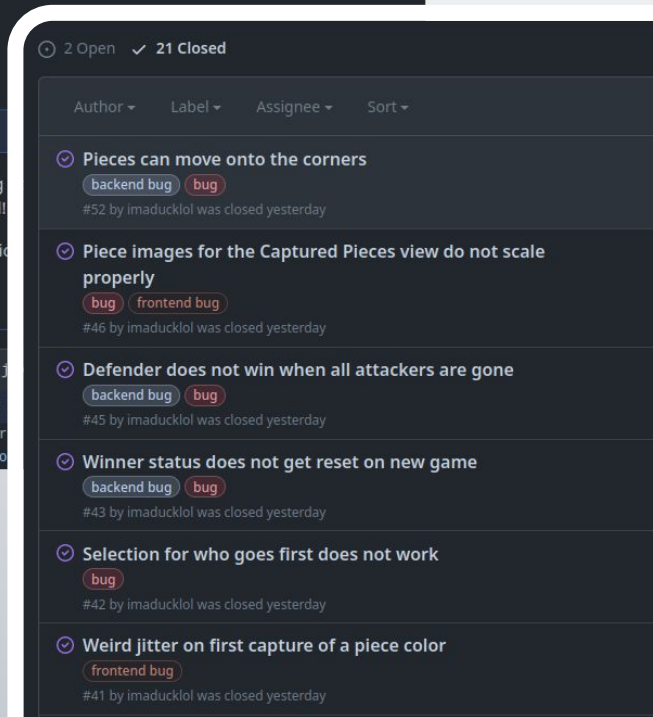
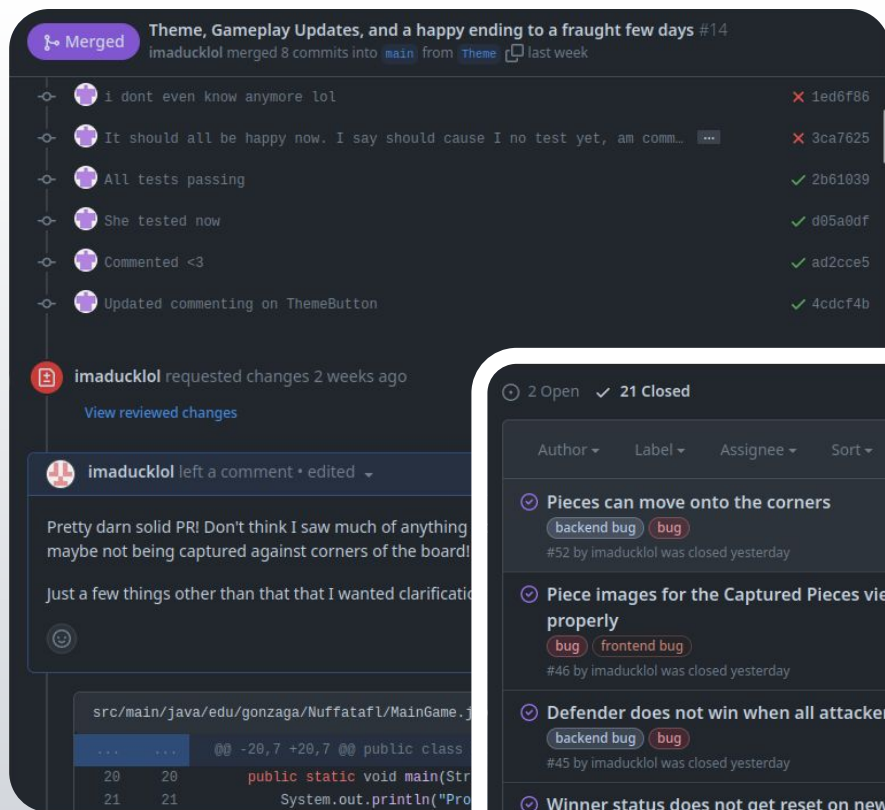
Frontend UML (Greatly Simplified)



Teamwork + Collaboration

We utilized a variety of services for team collaboration

- Discord
- GitHub Issues
- GitHub Pull Requests
- Code reviews
- In-person meetings



Testing + Validation + Acceptance

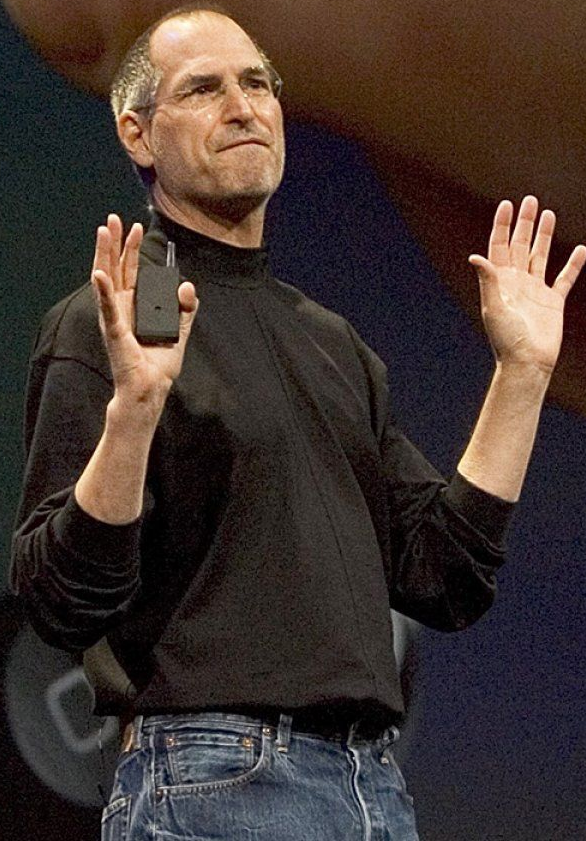
- All rules unit tested in multiple ways
 - 38 unit tests total for backend
 - Ensures confidence in working game logic
- All aspects of MVC tested
- User tests led to input changes
- 59 total unit tests

```
✓ BoardTest (edu.gonzaga.Nuffat 54 ms
  ✓ defenderWinOnLeft() 38 ms
  ✓ cannotMovePieceDiagonal() 1 ms
  ✓ piecelsCapturedWithCenter() 2 ms
  ✓ attackersDontWinSoloKing() 1 ms
  ✓ attackersWinWithCenter() 1 ms
  ✓ attackersDontWinKingSurrou 1 ms
  ✓ piecelsNotCaptured() 1 ms
  ✓ positionIsNotCornerSE() 1 ms
  ✓ cannotMovePieceOnOtherTea 1 ms
  ✓ cannotMovePieceToOccupiedTile()
  ✓ positionIsOnBoardMax() 1 ms
  ✓ positionIsOnBoardMin()
  ✓ successfullySwappedPieces()
  ✓ positionIsOffBoardMaxX()
  ✓ positionIsOffBoardMaxY()
  ✓ positionIsOffBoardMinX() 1 ms
  ✓ positionIsOffBoardMinY()
  ✓ canMovePieceToEdgeX()
  ✓ canMovePieceToEdgeY()
  ✓ canMovePieceToEmptyTile()
  ✓ piecelsCaptured()
  ✓ defenderWinOnRight() 1 ms
  ✓ defenderWinOnBottom()
  ✓ attackersDontWinMissingSoldier()
  ✓ positionIsOnCenter()

✓ Tests passed: 38 of 38 tests - 54 ms
C:\Users\cashh\.jdk\openjdk-21.0.2

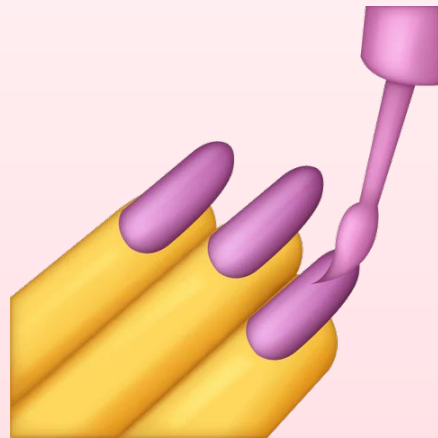
Process finished with exit code 0
```

Demo



The Slay Summary

- Git is awesome, we slay at it
- Observer and MVC are super powerful, also a slay
- Java Swing is NOT a slay
- Java Unit Testing? A slay for sure.
- Mark learned to have considerably sized PRs, now he slays at them



Bloopers

Orion, you're at 542 lines
Cash, you're at 630 lines
I'm at 2,152 🤔



markregg 04/17/2024 11:59 PM

Lmao I don't envy you

I've learned my lesson about small PRs I promise

April 18, 2024



imaducklol 04/18/2024 12:04 AM

I did it, it's done



(Not about the code, just the 700 lines)



Chill 04/23/2024 3:04 PM

how are we race conditioning without multithreading?



imaducklol 04/23/2024 3:05 PM

Good point, its not actually a race condition. Functions are just running in the order that I dont want them to



imaducklol 04/23/2024 3:12 PM

I was wrong! I jsut wasnt running the function, thats why it didnt exist



markregg Today at 5:40 PM

I'm on my way over

Any good options tonight?



imaducklol Today at 5:42 PM

Beef broth and brisket at simple servings, chicken taco salad bowl at 360, crispy chicken fritters at spikes



markregg Today at 5:42 PM

Woah a whole report. I gay thoughtfulness lol