P-16:
Code Structure
Leaderboard System Plan

March 5, 2025

1. Leaderboard (Main Controller)

Purpose: Central coordinator for leaderboard operations.

- Handles user requests and updates the leaderboard accordingly.

Attributes:

- database: LeaderboardDatabaseStub
- sorter: LeaderboardSorter
- updater: LeaderboardUpdater
- gui: LeaderboardGUI

Methods:

- initialize(): void - Links GUI, database, sorter.
- updateScores(matches: List): void - Triggers updater for new results.
- getTopPlayers(count: int): List - Fetches sorted data from sorter.
- exportLeaderboard(filePath: String): void - Exports to file.

Data Flow:

1. Calls LeaderboardUpdater when new match results arrive.
2. Retrieves sorted data via LeaderboardSorter.
3. Sends data to LeaderboardGUI for display.

2. LeaderboardGUI

Purpose: Displays the leaderboard UI.

- Allows users to view and interact with the leaderboard, users can search, filter, view player profiles, etc.

Attributes:

- searchBar: JTextField
- filterButton: JButton
- refreshButton: JButton
- leaderboardPanel: JPanel

Methods:

- displayLeaderboard(players: List): void - Renders data.
- filterLeaderboard(criteria: String): void - Applies filters.
- refresh(): void - Requests updated data from controller.
- search(query: String): void - Filters by name/ID.

Data Flow:

1. Requests sorted data from LeaderboardSorter.
2. Displays results to the player.

3. LeaderboardSorter

Purpose: Sorts players based on criteria

Attributes:

- strategy: SortCriteria - Sorting logic.
- players: List - Data to sort.

Methods:

- setSortStrategy(strategy: SortCriteria): void - Sets criteria: score, wins, etc.
- sortPlayers(): List - Returns sorted list.

Data Flow:

1. Fetches raw data from LeaderboardDatabaseStub.
2. Sorts using the active SortCriteria.
3. Returns sorted data to Leaderboard and GUI.

4. SortCriteria

Purpose: Defines how to sort (e.g., score, ascending, descending, win ratio, loss, etc).

Attributes:

- order: String - ascending/descending.
- criteria: String - score, wins, losses.

Methods:

- compare(p1: Player, p2: Player): int - Core comparison logic.
- setOrder(order: String): void - Ascending/descending.
- setCriteria(criteria: String): void - Score, wins, etc.

Default:

- criteria = "wins"
- order = "descending"

5. LeaderboardUpdater

Purpose: Updates the database with new match result

Attributes:

- matchResults: List

Methods:

- processMatchResult(match: Match): void - Updates stats for all players in the match.
- updatePlayerStats(player: Player, result: Result): void - Modifies wins, losses, score.
- calculateWinRatio(player: Player): float - Computes win/loss ratio.

Data Flow:

1. Receives match results.
2. Updates LeaderboardDatabaseStub with new stats.

6. LeaderboardDatabaseStub

Purpose: Temporary data storage for players and stats.

7. Overall System Flow

1. **Game Ends** → Match results generated.
2. LeaderboardUpdater processes results → updates LeaderboardDatabaseStub.
3. Leaderboard calls LeaderboardSorter → sorts data using SortCriteria.
4. LeaderboardGUI displays the sorted rankings.