# Project 6 Update

Project Details

Title: Love Island Game Show Simulator

Team: Alexis Davidson, Tiara Gibson, Khushi Jani

Work Done

Khushi's Work Progress:

- Implemented Contestant interface to serve as a base for all contestants.
- Used Contestant interface in implementation of Islander and RoleDecorator.
- Implemented concrete LeaderRole, FlirtRole, OutsiderRole, and DoubleFacedRole subclasses of RoleDecorator.
- Designed BehaviorStrategy interface to define contestant behaviors and some concrete subclasses for AggressiveStrategy and SafeStrategy.
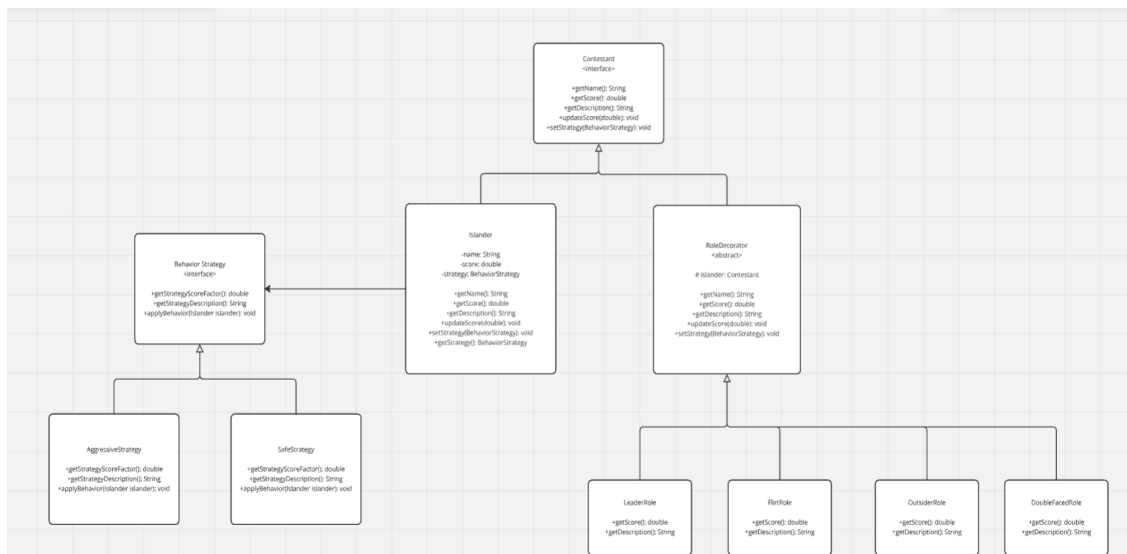
Tiara's Work Progress:

- Created ChallengeCommand interface with execute method to define different challenges.
- Implemented three challenge types: TriviaChallenge, PhysicalChallenge, and SocialChallenge, each assigning random scores to participants.

- Defined IslanderBehaviorStrategy interface for different approaches to challenges.
- Developed three strategies: AggressiveStrategy, BalancedStrategy, and PassiveStrategy, each influencing how the Islander interacts with challenges.
- Implemented ScoreTrackerSingleton to store and update islander scores in a centralized, singleton data store.
- Used updateScore method in ScoreTrackerSingleton to track scores for each Islander and retrieve score data.
- Created MainApp class with basic setup for JavaFX, initializing Islander objects with specific strategies and executing challenges.
- Displayed current scores in the console using the ScoreTrackerSingleton.
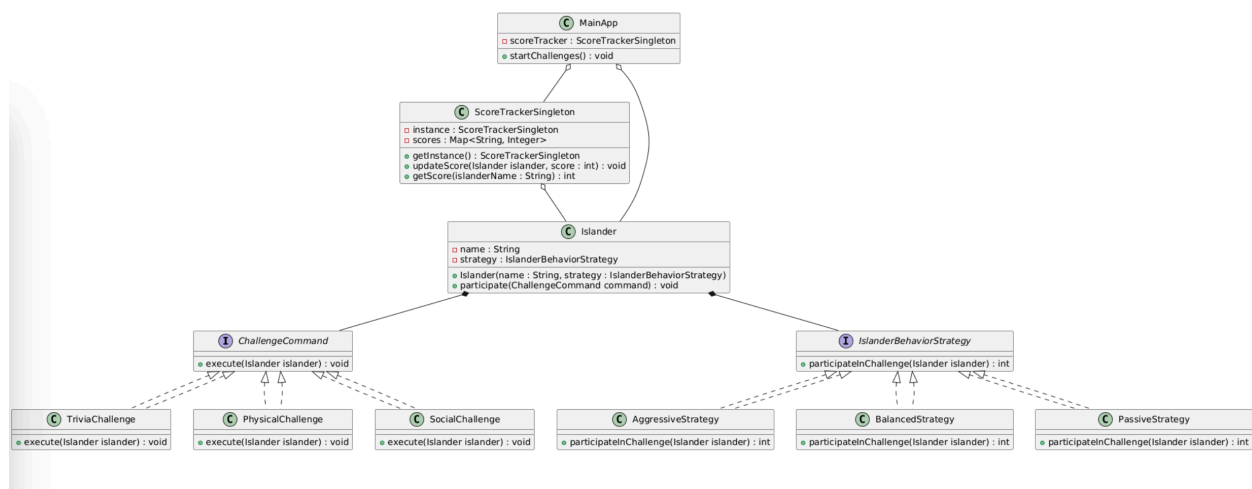
Alexis' Work Progress:

- Created the initial Java/JavaFX project and the GitHub repository.
- Completed the overall setup for the file structure for the UI of the application.
- Created all initial views and set up button actions across entire application, allowing the navigation from the start of the application through to the end of the game.
- Created the islander setup page, allowing for dynamic content to be populated on load.

Changes or Issues Encountered

Has anything changed so far in your approach to the project from the initial design in Project 5?

- Initial project creation was a little more complex than expected to get JavaFX working on individual machine.
- One change we've come across is the potential addition of enums in the project that we didn't account for in the planning phase (i.e. for behavior strategies or roles).

Patterns

- Command and Strategy give us flexibility in adding new challenges and behaviors.
- Singleton centralizes score management, reducing errors and ensuring consistency.
- Decorator and Observer patterns enable more customization and responsiveness in the UI.

Work To Do

- Design and implement JavaFX reporting screen to display final scores and challenge logs.
- Validate compatibility of Decorator roles, Command challenges, and Strategy behaviors with scoring system.
- Complete all connections from backend simulation functionality to the UI to ensure all strategies/roles can be accessed by the user.
- Establish error handling for edge cases that may be reached when users interact with the UI.