Changes from Part 1 to Part 3

This document outlines the changes made from Part 1 to Part 3 of the project, based on grading comments and feedback. The improvements address specific issues related to the **Domain Model**, **System Sequence Diagram**, **Behavior Contract**, and **Build and Test Automation**.

Domain Model

Feedback

- **Incorrect Attribute Types**: The initial domain model included types for attributes (e.g., `color: string`), which is not standard in UML domain models. Attributes should be shown without types to maintain a high-level conceptual focus.

Improvements

- **Removed Types from Attributes**: In Part 3, types were removed from attributes. This aligns with UML conventions for domain models, where attributes like `color`, `height`, or `material` are shown without types to represent game concepts abstractly.

System Sequence Model

Feedback

1. **Wrong Level of Abstraction**: The initial system sequence diagram included too many internal system interactions. It should have only displayed interactions between the **Player(s)** and the **Game System**.

2. **Incorrect Call Direction**: The model incorrectly included calls from the system to the user (e.g., `playerWins()`). Calls in sequence diagrams should originate from the user (Player) to the system, with the system only responding.

3. **Incomplete Modeling**: The initial model missed crucial parts of the game flow, such as initialization, player moves, building actions, and winner reporting.

Improvements

1. **Refined Level of Abstraction**: The updated sequence diagram removes internal system actions and focuses on high-level interactions between the **Player(s)** and **Game System**.

2. **Corrected Call Direction**: All calls now correctly originate from the user to the system. Actions like `takeTurn()`, `moveWorker()`, and `buildStructure()` flow from the player to the system, ensuring the user initiates each interaction.

3. **Added Missing Elements**: The model now includes:

    - **Game Initialization**: The player initiates the game setup.
    - **Turn-Based Actions**: Each turn includes player actions for moving and building.

  - **Win Notification**: The system notifies the player with `playerWins()` when the win condition is met, followed by `reportWinner()` to conclude the game.

These changes ensure the sequence model is complete and correctly focused on user-system interactions.

Behavior Contract (Precondition for Complete Tower Check)

Feedback

- **Missing Complete Tower Precondition**: The behavior contract initially lacked a precondition to check if a tower is "complete" (i.e., has three levels with a dome on top).

Improvements

- **Added Complete Tower Precondition**: The updated contract includes a precondition that checks both the tower level (must be level 3) and the presence of a dome on top. This ensures complete towers are tracked correctly, preventing further building and adhering to game rules.

Build and Test Automation

Feedback

- **Lack of Test Execution**: Although the project used GitHub Actions for continuous integration, it did not execute tests.

Improvements

- **Added Test Execution in CI**: The GitHub Actions workflow was modified to include test execution. Now, each build run also executes available tests, allowing for automated verification of code functionality and correctness.

Summary of Improvements

The changes made from Part 1 to Part 3 address all feedback by:

- **Improving the Domain Model** to remove types from attributes.
- **Refining the System Sequence Diagram** to focus on user-system interactions, add necessary actions, and correct call directions.
- **Enhancing the Behavior Contract** with a complete tower precondition.
- **Expanding Build and Test Automation** to execute tests in GitHub Actions.