

EndGame Phase Sequence Diagram Explanation

June, 2024

1 Introduction

This document provides a formal explanation of the sequence diagram representing the flow of method calls in the end game phase of a match. The controller manages various phases of a turn, and upon entering the endGameState, the `matchEnded()` method is called. This method invokes another method in the `Game` class to calculate and update player scores.

2 Flow of Method Calls

The sequence diagram illustrates the following sequence of method calls:

1. The `Controller` calls `matchEnded()` on the `EndGameState` object.
2. The `EndGameState` calls `getPlayers()` on the `Game` object to retrieve the list of players.
3. For each player, the `setStrategies()` method is called:
 - The `EndGameState` retrieves the strategy map from the `Game` object and initializes it for the player.
 - The `gatherPatterns()` and `gatherTotem()` methods collect the respective objective cards for the player.
 - The gathered cards are processed:
 - If the card ID starts with "OR", a `ConcreteOR` strategy is created.
 - If the card ID starts with "OP" and the shape is "STAIRS", a `ConcreteStair` strategy is created.
 - If the card ID starts with "OP" and the shape is "CHAIR", a `ConcreteChair` strategy is created.
 - If the card does not match any of the above conditions, an `IllegalCommandException` is thrown.
4. The `getPatternsTotemPoints()` method is called on the `Game` object to calculate the points for each player based on their strategies (virtual points).
5. If new points are greater than old points, the player's satisfied objectives are updated (satisfiedObj is needed to break ties).
6. The visited attribute of each card's coordinate is reset in order not to consider patterns which share common cards.
7. The `updateActualScore()` method is called on the `Board` object to update the player's actual score.