**Performance Optimization**

* **Caching Mechanism Implementation:**
  + Added three cache maps (scoreCache, worstCaseCache, and averageCaseCache) to the Corpus class.
  + These caches avoid repeated calculations and significantly improve the efficiency of the scoring functions.
* **Parallel Computation:**
  + Leveraged parallel stream processing in the scoreWorstCase and scoreAverageCase methods.
  + Utilized the parallel() method to fully exploit multi-core CPU performance.

**Enhanced Error Handling**

* **Custom Exception Framework:**
  + Introduced more specific exception types, such as EmptyCorpusException.
  + Improved clarity and readability of error messages.

**User Interface Improvements**

* **Visual Theme Support:**
  + Added a dark mode toggle feature.
  + Defined dedicated color constants and theme management methods.
* **Difficulty Levels:**
  + Implemented a game difficulty selector supporting Easy, Medium, and Hard modes.
  + Each difficulty level provides different degrees of hints.
* **Keyboard Shortcuts:**
  + Added keyboard shortcuts such as pressing Enter to submit guesses and Ctrl+N to start a new game.
  + Boosted user operational efficiency.

**Game Feature Extensions**

* **Game State Management:**
  + Implemented saving and loading of game state.
  + Supported resuming the game after interruption.
* **UI Initialization Fixes:**
  + Correctly initialized the feedbackArea component, resolving the previous null pointer exception.

**Code Structure Optimization**

* **Modular Design:**
  + Divided functionalities into clear methods such as toggleDarkMode and adjustDifficulty.
  + Improved code readability and maintainability.

### Log

1. **Removed Unused Methods**
   * Deleted unused methods checkGameTermination(), handleSingleCandidate(), and handleEmptyCorpus() from MatchleGUI. These methods only printed information to the console rather than updating the GUI.
2. **Removed Unused Constants**
   * Removed the FALSE constant from Filter, replacing it with direct filter creation wherever needed.
   * Replaced references to Filter.FALSE in NGramMatcher by directly creating filters.
3. **Optimized Code Structure**
   * Refactored the checkGameTermination() method in MatchleGame and extracted sub-methods to reduce complexity.
   * Removed the main and testHardCase methods from CorpusLoader, as they were only used for testing and not needed in production code.
4. **Unified Exception Handling**
   * Consolidated serialVersionUID management in exception classes by defining it only in the top-level exception class.
5. **Created a Shared Game Logic Class**
   * Extracted a GameLogic class containing shared game logic for both MatchleGame and MatchleGUI:
     + Game state management
     + Filter handling
     + Candidate word management
     + Game termination checks
6. **Unified Object Creation Pattern**
   * Standardized the factory method pattern in the Filter class, making its constructor private.