Implementation Manual for SilentSymbols Application

Application Overview

The SilentSymbols application is designed to provide an interactive game experience that challenges users to identify letters and words. The application leverages different skill levels, allowing players to choose between easy and hard modes. The implementation is structured around several classes that manage the game's core functionality.

Class Structure

1. SilentSymbols Class

This is the main class responsible for initiating and managing the application.

Attributes:
- `static final double IMAGE_WIDTH`
- `static final double IMAGE_HEIGHT`
- `HashMap<String, String> wordBank`
- `List<String> letterBank`
- `Random random`
- `String currentWord`
- `String currentLetter`

Methods:
- `void start(Stage primaryStage)`: Initializes the game and sets up the primary stage for the UI.
- `void showEasyScene(Stage primaryStage)`: Displays the easy difficulty level.
- `void showHardScene(Stage primaryStage)`: Displays the hard difficulty level.
- `Scene createMainScene(Stage primaryStage)`: Creates the main scene layout.
- `HashMap<String, String> loadWordBank()`: Loads the word bank from a predefined source.
- `List<String> loadLetterBank()`: Loads a list of letters for the game.
- `static void main(String[] args)`: Main entry point of the application.

2. Images Class

This class manages image assets used in the game.

- Attributes:
  - `String fileName`: The name of the image file.

- Methods:
  - `Image(String fileName)`: Constructor that initializes the image.

3. Easy Class

This class represents the easy level of the game.

- Attributes:
  - `String answer(String letterAnswer)`: Method to handle user answers at the easy level.

4. Hard Class

This class represents the hard level of the game.

- Attributes:
  - `String answer(String wordAnswer)`: Method to process user responses at the hard level.

5. LetterValidity Class

This class is responsible for checking the validity of letter answers.

- Methods:
  - `Check(String userAnswer)`: Checks if the user-provided answer is valid.

6. WordValidity Class

This class checks the validity of word answers.

- Methods:
  - `Check(String userAnswer)`: Determines the correctness of the user's word answer.

Design Considerations
- Game Flow: The application begins with the `start` method in the `SilentSymbols` class, which prompts the user to choose a difficulty level and subsequently displays the corresponding scene.

- Data Management: The application uses `HashMap` for storing the word bank and `List` for managing the letter bank, allowing for efficient retrieval and storage.

- User Interaction: The `Easy` and `Hard` classes include methods that capture user inputs and respond accordingly, thereby enhancing interactivity.

Conclusion

The SilentSymbols implementation follows a structured approach to the design and functionality of the application. With clear delineation of responsibilities among various classes, the application is poised to deliver an engaging user experience while maintaining maintainability and scalability. Future enhancements could include adding multiplayer features or additional difficulty levels.