**Word Guessing Game Developer's Guide**

Welcome to the Developer's Guide for the "Word Guessing Game" project. This guide will provide a detailed description of the modules and code structure, making it easier for other developers to understand and contribute to the project.

**Table of Contents**

1. Project Structure
2. Main Functionality
3. Module Descriptions
   * Word Selection Module
   * Guess Validation Module
   * Feedback Generation Module
   * Game Loop Module

**1. Project Structure**

The "Word Guessing Game" project is organized into distinct modules to improve code readability and maintainability. Below is an overview of the project structure:

* **Main Script (word\_guessing\_game.py):** This script contains the main game logic, including user interaction and game flow control.
* **Word Lists (word\_lists.py):** This module stores lists of words categorized by difficulty level (beginners, moderate, expert).
* **Modules (Directory):** This directory contains separate Python files, each representing a specific module or functionality of the game.

**2. Main Functionality**

The main functionality of the "Word Guessing Game" involves:

* Selecting a word based on the chosen difficulty level.
* Validating user input for guess format (alphabets only and correct length).
* Comparing the user's guess with the secret word to provide feedback.
* Implementing a game loop that allows users to make multiple guesses within a set number of attempts.
* Ending the game when the user either guesses the word correctly or runs out of attempts.

**3. Module Descriptions**

Let's dive into detailed descriptions of the key modules that make up the "Word Guessing Game" project:

Word Selection Module <a name="word-selection-module"></a>

* **Module Name:** **word\_lists.py**
* **Description:** This module stores lists of words categorized by difficulty level. The appropriate word list is selected based on the user's chosen difficulty level.
* **Purpose:** To provide a source of words for the game, making it easy to add, modify, or expand word lists for different levels.

Guess Validation Module <a name="guess-validation-module"></a>

* **Module Name:** **guess\_validation.py**
* **Description:** This module is responsible for validating user input for guess format. It checks if the input consists of alphabets only and matches the correct length.
* **Purpose:** To ensure that user inputs are valid and meet the game's requirements.

Feedback Generation Module <a name="feedback-generation-module"></a>

* **Module Name:** **feedback\_generation.py**
* **Description:** This module generates feedback on the user's guess by comparing it with the secret word. It determines which letters are correct and in the correct position and which letters are incorrect.
* **Purpose:** To provide meaningful feedback to the user after each guess, helping them narrow down the correct word.

Game Loop Module <a name="game-loop-module"></a>

* **Module Name:** **game\_loop.py**
* **Description:** This module implements the core game loop. It allows users to make guesses within a set number of attempts, checks for game-ending conditions (correct guess or out of attempts), and controls the flow of the game.
* **Purpose:** To manage the gameplay, including attempts, user input, and game termination conditions.

Each module is designed to be modular and independent, allowing for easy testing and future enhancements. Developers can work on specific modules or extend the game's functionality with minimal impact on other parts of the code.

By understanding the structure and purpose of these modules, developers can contribute to the project, add new features, or improve existing ones while maintaining code consistency and readability.