Report: Analysis of DoodleJava Code

1. Introduction

The purpose of this report is to analyze the provided code in the DoodleJava package. The code represents a simple game called Doodle Jump, implemented using Java and Swing. The game is displayed in a JFrame and involves a player-controlled character jumping on platforms to reach higher levels.

2. Code Overview

The code consists of two classes: `Frame` and `SwapPanelListener`.

2.1 Frame Class

The `Frame` class extends the `JFrame` class and serves as the main window for the game. It contains the game panel and handles user interaction.

2.1.1 Constants and Initialization

The `Frame` class defines a constant `SIZE` that represents the dimension of the game window. The value of `SIZE` is calculated based on the screen height, ensuring that the game window fits within the screen while maintaining a fixed width of 500 pixels.

2.1.2 Constructor

The constructor of the `Frame` class initializes the JFrame by setting various properties. It sets the window's icon image, sets the default close operation to exit the application when the window is closed, and makes the window non-resizable. The window is then packed, and its size is set to the calculated `SIZE`. The window is centered on the screen using `setLocationRelativeTo(null)`.

2.1.3 Look and Feel

The `Frame` class attempts to set the look and feel of the application to the system's default look and feel using `UIManager.setLookAndFeel()`. Any exceptions that occur during this process are caught and printed to the standard error stream.

2.1.4 Game Panel

An instance of the `GamePanel` class is created and added to the `Frame` using `this.add(this.gamePanel)`. The `GamePanel` is a custom JPanel that represents the game area.

2.1.5 Event Listeners

The `Frame` class sets up event listeners for the `GamePanel` by implementing the `NewGameListener` interface. These listeners handle updates to the game's score and the game-over state. The `setTitle()` method is called to update the window's title with the current score.

2.1.6 Start Method

The `start()` method displays a JOptionPane dialog to prompt the user with the message "Ready...?" and an option to start the game. Upon clicking the "GO!" button, the `start()` method calls the `start()` method of the `GamePanel` to begin the game.

2.1.7 Main Method

The `main()` method creates an instance of the `Frame` class, sets it visible, and calls the `start()` method to start the game.

2.1.8 Helper Methods

The `Frame` class also contains two helper methods:

- `setTitle(int points)` sets the window's title to display the current score.

- `setSizeWithoutInsets(Dimension d)` adjusts the window's size to include the insets (borders) by calculating the total width and height.

2.2 SwapPanelListener Class

The `SwapPanelListener` class implements the `ActionListener` interface and is used to swap between different panels in the game.

2.2.1 Constructor

The constructor of `SwapPanelListener` takes two JPanel parameters: `pFrom` and `pTo`. These parameters represent the panels that need to be swapped.

2.2.2 actionPerformed Method

The `actionPerformed()` method is triggered when an action event occurs, such as a button click. It sets the visibility of `pFrom` to false and `pTo` to true, effectively swapping the panels.

3. Conclusion

In conclusion, the provided code represents a simple implementation of the Doodle Jump game using Java and Swing. The `Frame` class serves as the main window for the game, handling user interaction and displaying the game panel. The `SwapPanelListener` class is used to swap between different panels in the game. The code provides a starting point for further development and customization of the Doodle Jump game.