Swipe Input Documentation

This documentation explains the functionality and usage of a swipe gesture recognition system implemented in Unity3D. The system allows users to perform various actions by swiping in different directions - up, down, left, and right. The system consists of two scripts: `SwipeInput` (the core module) and `SwipeInputController` (the usage script).

SwipeInput Script

Core Variables

- `tapPositions`: An array of Vector2 to store the positions of two consecutive taps.
- `swipePositions`: An array of Vector2 to store the positions of two consecutive swipes.
- `offsetTap`: A float representing the allowed offset for a tap gesture.
- `offsetSwipe`: A float representing the allowed offset for a swipe gesture.
- `fTapAllowed`: A boolean flag indicating whether tap gestures are allowed.
- `fSwipeAllowed`: A boolean flag indicating whether swipe gestures are allowed.
- `tempX` and `tempY`: Temporary variables to store the differences in x and y coordinates during gesture processing.

Core Methods

- `ProcessTouches()`: Handles the touch input and updates the tap and swipe positions based on touch phases.
- `ResetPositions()`: Resets tap and swipe positions and flags.

Control Methods

- `Tap()`: Detects tap gestures by comparing the distance between two tap positions with the specified offset.
- `SwipeLeft()`, `SwipeRight()`, `SwipeUp()`, `SwipeDown()`: Detects swipe gestures in the respective directions by comparing the distance between two swipe positions with the specified offset.

SwipeInputController Script

Variables

- `Debug`: A reference to a Text component for displaying debug information.

Methods

- `Start()`: Initializes the `Debug` variable by finding the Text component.

- `Update()`: Processes touch input using `SwipeInput.ProcessTouches()` and performs actions based on detected gestures. Updates the object's position accordingly.

Usage

1. Attach the `SwipeInputController` script to the GameObject you want to control with swipe gestures.
2. Create a UI Text object named "Debug" to display debug information.
3. The system automatically detects swipe gestures and updates the object's position based on the detected gesture.

Example Usage

```
Vector3 pos = this.transform.position;

SwipeInput.ProcessTouches();

if (SwipeInput.Tap()) {
    // Handle tap gesture if needed
} else if (SwipeInput.SwipeUp()) {
    pos.z += 1;
    Debug.text = "Swipe Up";
} else if (SwipeInput.SwipeDown()) {
    pos.z -= 1;
    Debug.text = "Swipe Down";
} else if (SwipeInput.SwipeLeft()) {
    pos.x -= 1;
    Debug.text = "Swipe Left";
} else if (SwipeInput.SwipeRight()) {
    pos.x += 1;
    Debug.text = "Swipe Right";
}

this.transform.position = pos;
```