Menu Button Notification System - Usage Guide

Overview

The refactored menu button system provides automatic notifications that:

- Show "NEW!" badge when buttons are first unlocked
- Display persistent notifications until the button is clicked
- Support multiple notification types (alerts, info, counts)
- Auto-clear when the button is clicked

Notification Types

1. NEW Badge (Automatic on Unlock)

Appearance: Green "NEW!" badge in bottom-right corner **When:** Automatically shown when `unlock()` is called on a locked button **Clears:** When user clicks the button

```
java

// Automatically shows NEW badge
uiManager.unlockMenuButton("inventory");
```

2. Alert Notification (!)

Appearance: Red pulsing "!" in top-right corner **Use Cases:** Urgent updates, level ups, important items **Clears:** When user clicks the button

```
java

// Level up notification
uiManager.notifyLevelUp();

// Inventory item added
uiManager.notifyInventoryUpdate();
```

3. Info Notification (?)

Appearance: Yellow pulsing "?" in top-right corner **Use Cases:** Quest completions, optional updates **Clears:** When user clicks the button

```java
// Quest completed
uiManager.notifyQuestComplete();
```

4. Count Badge

Appearance: Red number badge in top-right corner **Use Cases:** Number of active quests, unread messages **Clears:** When user clicks the button

```java
// Show number of active quests
uiManager.notifyQuestUpdate(3);  // Shows "3"

// Clear count
uiManager.notifyQuestUpdate(0);  // No badge
```

Common Use Cases

First-Time Unlock Flow

```java
// 1. Button starts locked
UIButton inventoryBtn = uiManager.getMenuButton("inventory");
// inventoryBtn.isLocked() == true

// 2. Unlock it (automatically shows NEW badge)
uiManager.unlockMenuButton("inventory");
// Shows green "NEW!" badge

// 3. User clicks button
inventoryBtn.onClick();
// NEW badge automatically cleared
```

Item Added to Inventory

```java
```

```
// Add item to inventory
Item sword = ItemManager.createWoodenShortSword();
uiManager.addItemToInventory(sword);

// Inventory button now has red "!" notification
// Cleared when user opens inventory
```

Quest System Integration

```
java

// Quest accepted
uiManager.unlockQuestButton();
// Shows "NEW!" badge + count "1"

// Quest completed
uiManager.notifyQuestComplete();
// Shows yellow "?"

// Check active quests
QuestLog questLog = player.getComponent(QuestLog.class);
int activeCount = questLog.getActiveQuestCount();
uiManager.notifyQuestUpdate(activeCount);
// Shows count badge with number
```

Level Up Notification

```
java

// Player levels up
private void handleLevelUp(Entity player) {
    Experience exp = player.getComponent(Experience.class);
    Stats stats = player.getComponent(Stats.class);

    // Apply level up stats
    stats.applyLevelStats(exp, true);

    // Notify UI
    uiManager.notifyLevelUp();
    // Stats button shows red "!" notification
}
```

Manual Notification Control

Set Custom Notifications

```java
UIButton button = uiManager.getMenuButton("stats");

// Set alert
button.setNotification(UIButton.NotificationType.ALERT);

// Set info
button.setNotification(UIButton.NotificationType.INFO);

// Set count
button.setCountNotification(5);

// Set custom text
button.setNotification(UIButton.NotificationType.COUNT, "99+");
```

Clear Notifications

```java
// Clear specific button
UIButton button = uiManager.getMenuButton("quest");
button.clearNotification();

// Clear all buttons (debugging)
uiManager.clearAllNotifications();
```

Check Notification Status

```java
```

```java
UIButton button = uiManager.getMenuButton("inventory");

// Check if has notification
if (button.hasNotification()) {
    // Get type
    UIButton.NotificationType type = button.getNotificationType();

    // Check if was just unlocked
    if (button.wasJustUnlocked()) {
        System.out.println("Button was just unlocked!");
    }
}

// Debug: Print all notification statuses
uiManager.printNotificationStatus();
```

Integration Examples

Example 1: Complete Quest Flow

java

```
// Quest accepted (in handleQuestAccept)
Quest quest = dialogueBox.getOfferedQuest();
if (quest != null) {
    quest.accept();

    QuestLog questLog = player.getComponent(QuestLog.class);
    if (questLog != null) {
        questLog.addQuest(quest);
    }

    // Unlock quest button with NEW badge
    uiManager.unlockQuestButton();

    // Update quest count
    uiManager.updateQuestIndicator();
}

// Quest completed (in updateQuestProgress)
if (quest.isCompleted()) {
    // Show completion notification
    uiManager.notifyQuestComplete();
}
```

Example 2: Inventory System

java

```
// Add item
public boolean addItemToInventory(Item item) {
    if (inventoryGrid.addItemToCurrentTab(item)) {
        // Show notification
        uiManager.notifyInventoryUpdate();
        return true;
    }
    return false;
}

// User opens inventory
private void toggleInventory() {
    inventoryContainer.setVisible(!inventoryContainer.isVisible());
    // Notification automatically cleared by button onClick()
}
```

Example 3: Stats Update

```java
// Player levels up
if (levelsGained > 0) {
    // Update stats
    stats.applyLevelStats(exp, true);

    // Award skill points
    if (skillLevel != null) {
        exp.awardSkillPoints(levelsGained, skillLevel);
    }

    // Notify user
    uiManager.notifyLevelUp();
    // Red "!" appears on stats button
}

// User opens stats panel
public void toggleStatsPanel() {
    isStatsVisible = !isStatsVisible;
    statsPanel.setVisible(isStatsVisible);
    // Notification automatically cleared
}
```

Best Practices

1. Use Appropriate Notification Types

- **NEW Badge:** Reserved for first unlock only (automatic)
- **Alert (!):** Important, time-sensitive updates
- **Info (?):** Optional updates, completions
- **Count:** Show quantity (quests, items, messages)

2. Let Auto-Clear Handle Cleanup

```java
```

```
// ✅ GOOD: Notification clears automatically
uiManager.notifyLevelUp();
// User clicks stats button → notification gone

// ❌ BAD: Manual clearing (unnecessary)
uiManager.notifyLevelUp();
statsButton.onClick();
statsButton.clearNotification(); // Not needed!
```

3. Update Counts Dynamically

```
java

// Update quest count whenever quests change
private void updateQuestIndicator() {
    QuestLog questLog = player.getComponent(QuestLog.class);
    int activeCount = questLog.getActiveQuestCount();

    // Shows count or hides if 0
    uiManager.notifyQuestUpdate(activeCount);
}
```

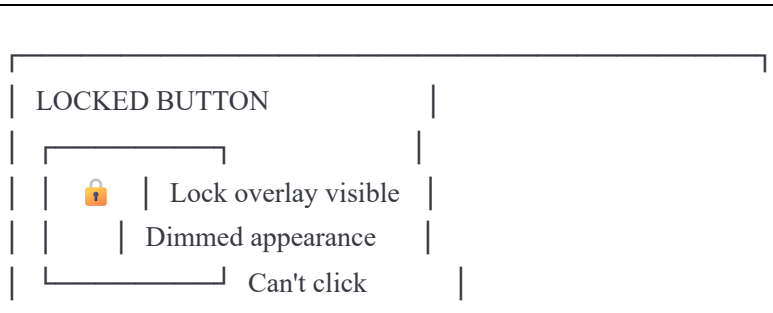
4. Combine Notifications When Appropriate

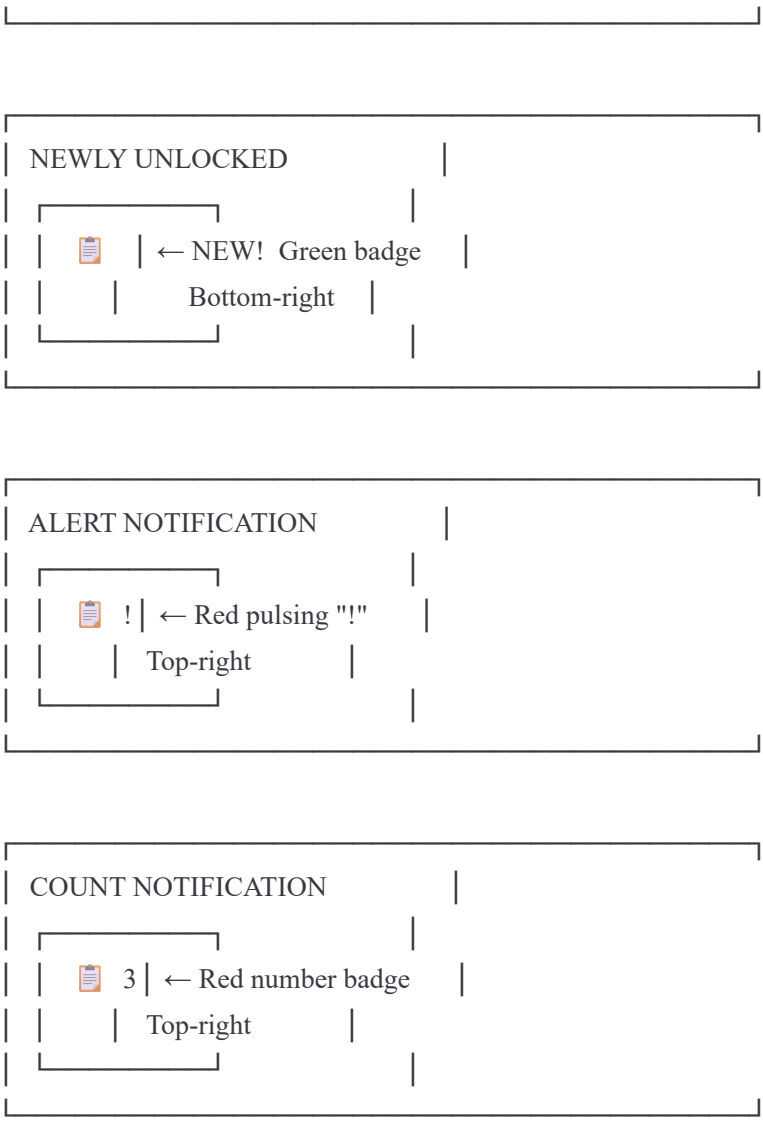
```
java

// Quest button can show both NEW and count
uiManager.unlockQuestButton();
// Shows: NEW badge + count "1"

// After clicking, NEW clears but count remains
questButton.onClick();
// Shows: count "1" only
```

Visual Reference





Debugging

Print Notification Status

java

```
// In game loop or debug command
if (debugMode) {
    uiManager.printNotificationStatus();
}

// Output:
// === Button Notification Status ===
// Settings: NONE
// Inventory: ALERT
// Stats: ALERT
// Quest: COUNT
// ...
// ==================================
```

Clear All for Testing

```
java

// Reset all notifications
uiManager.clearAllNotifications();
```

Migration from Old System

Old Code

```
java

// Old notification system
button.setNotification(true);  // Generic boolean
```

New Code

```
java

// New typed notification system
button.setAlertNotification();     // Specific type
button.setInfoNotification();      // Different visual
button.setCountNotification(3);    // Shows number
```

Benefits

- ✅ Visual variety (!, ?, numbers, NEW)

- ✅ Automatic clearing on click
- ✅ Auto-NEW badge on unlock
- ✅ Type-safe notification system
- ✅ Better user feedback