

# Rest System Setup Guide

## Overview

The resting system allows players to make camp after moving through the dungeon (when no encounter is triggered). During rest, all party members gradually heal 1 HP per second until fully restored, and the day counter advances.

## Files Created/Modified

### New Files:

1. **RestManager.cs** - Manages the resting system
2. **RestManager.cs.meta** - Unity metadata

### Modified Files:

1. **GameUIManager.cs** - Added `SetRestMode()` method
2. **PlayerPartyController.cs** - Integrated rest system with movement

## Setup Instructions

### 1. Create RestManager GameObject

1. In your Unity scene hierarchy, create a new empty GameObject
2. Name it "RestManager"
3. Add the `RestManager` component to it
4. Assign the **Campfire Sprite** in the inspector (you'll need a campfire image)

### 2. Configure RestManager Settings

In the RestManager inspector:

- **Heal Per Second:** 1 (default - adjust as needed)
- **Campfire Sprite:** Assign your campfire image sprite

### 3. No Code Changes Required

The system automatically integrates with existing managers through singleton references.

## How It Works

### Player Experience Flow:

1. **Player moves** through dungeon (W/A/S/D/Q/E keys)
2. **No encounter triggers** (95% chance normally)

3. **Rest prompt appears:** "Day X - Press Return to make camp"







4. Player can either:

- Press **Return** to rest
- Continue moving (clears the prompt)

### **Resting Sequence:**

1. Dungeon camera **hides**
2. Campfire image **displays**
3. Text shows: "You are resting..."
4. **Healing begins:** +1 HP/second for all party members
5. **Continues until** all party members reach max HP
6. **Day counter increases** by 1
7. **Returns to dungeon** view - movement enabled

### **Key Features:**

-  Rest offer only appears once per movement
-  Can't rest during combat
-  Movement clears rest offer without resting
-  All party members heal simultaneously
-  Day counter tracks progression
-  Smooth transitions between modes

## **Integration Points**

### **With CombatManager:**

- Rest system checks `CombatManager.Instance.IsInCombat`
- Cannot rest during combat
- Cannot move during rest

### **With PlayerPartyController:**

- Movement triggers rest offers (when no encounter)
- Movement inputs clear rest offer
- Controller disabled during rest

## With GameUIManager:

- New `SetRestMode()` shows campfire
- Reuses existing encounter frame
- Returns to dungeon mode after rest

## Testing Checklist

- ☐ RestManager GameObject exists in scene
- ☐ Campfire sprite is assigned
- ☐ Move through dungeon - see rest prompt
- ☐ Press Return - see campfire and healing
- ☐ Verify HP increases each second
- ☐ Verify day counter increases after rest
- ☐ Test movement clears rest offer
- ☐ Test cannot rest during combat
- ☐ Test cannot move during rest

## Customization Options

### Healing Speed

```
csharp
public int healPerSecond = 1; // Change in inspector
```

### Rest Offer Text

In `RestManager.OfferRest()`:

```
csharp
GameUIManager.Instance.encounterText.text =
    $"Day {currentDay} - Press <u>Return</u> to make camp";
```

### Add Rest Bonuses

In `RestManager.EndRest()`, add:

```
csharp
```

*// Example: Restore mana, remove debuffs, etc.*

```
foreach (var player in partyMembers)
{
    player.RestoreMana();
    player.ClearDebuffs();
}
```

## Add Food/Supply Mechanics

Modify `StartRest()`:

```
csharp

if (GameInventory.Instance.HasFood())
{
    GameInventory.Instance.ConsumeFood();
    StartCoroutine(RestSequence());
}
else
{
    GameUIManager.Instance.encounterText.text =
        "No food to make camp!";
}
```

## Debug Tips

**Check Rest State:**

```
csharp

Debug.Log($"Is Resting: {RestManager.Instance.IsResting}");
Debug.Log($"Current Day: {RestManager.Instance.CurrentDay}");
Debug.Log($"Waiting for Input: {RestManager.Instance.WaitingForRestInput}");
```

**Force Rest (for testing):**

Add to RestManager:

```
csharp

[ContextMenu("Force Start Rest")]
void ForceRest()
{
    OfferRest();
    StartRest();
}
```

## Future Enhancements

### Possible Additions:

1. **Random Events** during rest (ambush chance)
2. **Rest Locations** (safe camps vs dangerous areas)
3. **Supply Management** (food required to rest)
4. **Character Interactions** during rest (dialogue)
5. **Skill Training** at camp
6. **Dream Sequences** with story elements
7. **Exhaustion System** (must rest after X moves)
8. **Time of Day** (day/night cycle)

## Troubleshooting

### Rest prompt doesn't appear:

- Check RestManager is in scene
- Verify encounter chance isn't 100%
- Ensure not in combat

### Can't press Return to rest:


- Check RestManager Update() is running
- Verify WaitingForRestInput is true
- Check keyboard input settings

### Healing doesn't work:

- Verify healPerSecond > 0
- Check PlayerCharacter.Heal() method
- Ensure party members are assigned

### Campfire doesn't show:

- Assign campfire sprite in inspector
  - Check encounterFrame is not null
  - Verify SetRestMode() is called
-

**System Status:**  Ready to use **Dependencies:** CombatManager, GameUIManager, PlayerPartyController  
**Unity Version:** Compatible with Unity 2020.3+