Quest Expansion Guide

Where to Add New Quests

You have **TWO quest systems** in your game:

1. IntroQuestHandler (Sequential Tutorial Quests)

- **Location**: `IntroQuestHandler.java`
- **Purpose**: Linear tutorial quests that unlock features
- **NPC**: Fionne only
- **Indicator**: Managed automatically by IntroQuestHandler

2. NPC Quest System (Traditional Quests)

- **Location**: `NPC.java` and `Quest.java`
- **Purpose**: Regular quests with objectives
- **NPCs**: Any NPC with quests added
- **Indicator**: Managed by NPC component (if GameLogic methods restored)

Option A: Add to IntroQuestHandler (Sequential)

Use this for **tutorial quests** that must happen in order.

Example: Add Stage 4 - Kill 5 Goblins

Step 1: Add Enum Values

In `IntroQuestHandler.java`, add to the `IntroStage` enum:

```
java
```

```
public enum IntroStage {
    NOT_STARTED,
    STAGE_1_DIALOGUE,
    STAGE_1_COMPLETE,
    STAGE_2_EQUIP_SWORD,
    STAGE_2_COMPLETE,
    STAGE_3_DIALOGUE,
    STAGE_3_COMPLETE,

    // ★ NEW: Stage 4
    STAGE_4_DIALOGUE,      // Fionne asks you to kill goblins
    STAGE_4_IN_PROGRESS,   // Killing goblins
    STAGE_4_COMPLETE,      // Return to Fionne for reward

    ALL_COMPLETE
}
```

Step 2: Add to handleFionneInteraction()

java

```
public boolean handleFionneInteraction(Entity npcEntity) {
    // ... existing code ...

    switch (currentStage) {
        // ... existing cases ...

        case STAGE_3_COMPLETE:
        case STAGE_4_DIALOGUE:
            showStage4Dialogue(ui);  // ★ NEW
            return true;

        case STAGE_4_IN_PROGRESS:
            checkStage4Progress(ui);  // ★ NEW
            return true;

        case STAGE_4_COMPLETE:
            showStage4Completion(ui);  // ★ NEW
            return true;

        case ALL_COMPLETE:
            // ... existing code ...
    }
}
```

Step 3: Add Dialogue Methods

java

```java
/**
 * ★ NEW: Stage 4 - Kill 5 Goblins
 */
private void showStage4Dialogue(UIManager ui) {
    UIDialogueBox dialogueBox = ui.getDialogueBox();

    currentStage = IntroStage.STAGE_4_DIALOGUE;

    dialogueBox.showMessageWithAccept(
        "The goblins on this island are becoming aggressive. Hunt 5 of them to prove your combat skills.",
        () -> {
            dialogueBox.showMessageWithAccept(
                "Be careful - they fight back!",
                () -> {
                    // Start the quest
                    currentStage = IntroStage.STAGE_4_IN_PROGRESS;
                    updateQuestIndicator();  // Change to "..."

                    // Track goblin kills (add field to IntroQuestHandler)
                    goblinsKilled = 0;

                    dialogueBox.setVisible(false);
                    System.out.println("[QUEST] Stage 4 started: Kill 5 Goblins");
                },
                () -> dialogueBox.setVisible(false)
            );
        },
        () -> {
            // Declined quest
            dialogueBox.setVisible(false);
        }
    );
}

/**
 * ★ NEW: Check progress during Stage 4
 */
private void checkStage4Progress(UIManager ui) {
    if (goblinsKilled >= 5) {
        // Quest complete!
        currentStage = IntroStage.STAGE_4_COMPLETE;
        updateQuestIndicator();  // Change to "?"
```

```
            ui.showDialogue(
                "Fionne",
                "You've defeated the goblins! Return to me for your reward."
            );
        } else {
            ui.showDialogue(
                "Fionne",
                "Goblins slain: " + goblinsKilled + " / 5. Keep hunting!"
            );
        }
    }

    /**
     * ★ NEW: Complete Stage 4 and give rewards
     */
    private void showStage4Completion(UIManager ui) {
        UIDialogueBox dialogueBox = ui.getDialogueBox();

        dialogueBox.showMessageWithAccept(
            "Well done! You've proven yourself in combat. Take this as a reward.",
            () -> {
                // Grant rewards
                Entity player = gameState.getPlayer();
                Experience exp = player.getComponent(Experience.class);

                if (exp != null) {
                    int levelsGained = exp.addExperience(500);

                    if (levelsGained > 0) {
                        Stats stats = player.getComponent(Stats.class);
                        if (stats != null) {
                            stats.applyLevelStats(exp, true);
                        }
                    }
                }

                // Add item reward
                ui.addItemToInventory(ItemManager.createHealthPotion());
                ui.notifyInventoryUpdate();

                // Advance to next stage or complete
                currentStage = IntroStage.ALL_COMPLETE;
                updateQuestIndicator();  // Hide indicator
```

```
            dialogueBox.setVisible(false);
            System.out.println("[QUEST] Stage 4 complete! All intro quests finished.");
        },
        () -> dialogueBox.setVisible(false)
    );
}
```

Step 4: Add Tracking Field

Add to IntroQuestHandler class:

java

```
// ★ NEW: Track Stage 4 progress
private int goblinsKilled = 0;

public void onGoblinKilled() {
    if (currentStage == IntroStage.STAGE_4_IN_PROGRESS) {
        goblinsKilled++;
        System.out.println("Goblin killed! (" + goblinsKilled + " / 5)");

        if (goblinsKilled >= 5) {
            currentStage = IntroStage.STAGE_4_COMPLETE;
            updateQuestIndicator();  // Change to "?"
            System.out.println("Quest complete! Return to Fionne.");
        }
    }
}
```

Step 5: Call from GameLogic

In `GameLogic.handleMonsterDeath()`:

java

```
private void handleMonsterDeath(Entity monster, Sprite sprite) {
    // ... existing code ...

    // ★ NEW: Notify intro quest handler
    if (monster.getName().equals("Goblin")) {
        IntroQuestHandler introHandler = state.getIntroQuestHandler();
        if (introHandler != null) {
            introHandler.onGoblinKilled();
        }
    }

    // ... rest of existing code ...
}
```

Step 6: Update Indicator Logic

In `updateQuestIndicator()`:

```
java

switch (currentStage) {
    // ... existing cases ...

    case STAGE_3_COMPLETE:
    case STAGE_4_DIALOGUE:
        indicator.show(IndicatorType.AVAILABLE);  // "!"
        break;

    case STAGE_4_IN_PROGRESS:
        indicator.show(IndicatorType.IN_PROGRESS);  // "..."
        break;

    case STAGE_4_COMPLETE:
        indicator.show(IndicatorType.COMPLETE);  // "?"
        break;

    // ... rest of cases ...
}
```

Option B: Add Traditional NPC Quest

Use this for **optional side quests** from any NPC.

Example: Add Quest to New NPC "Merchant"

Step 1: Create NPC in GameState

In `GameState.initializeWorld()`:

```
java

// Create Merchant NPC
Entity merchant = EntityFactory.createNPC("merchant", "Merchant Bob", 20 * 64, 10 * 64);
entities.add(merchant);
```

Step 2: Add Quest to NPC

In `EntityFactory.createNPC()` or separately:

```
java

NPC merchantComponent = merchant.getComponent(NPC.class);

Quest fetchQuest = new Quest(
    "fetch_wood",
    "Gather Wood",
    "The merchant needs 10 Carved Wood for repairs.",
    Quest.QuestType.COLLECT
);

fetchQuest.addObjective(new QuestObjective(
    "collect_carved_wood",
    "Collect 10 Carved Wood",
    10
));

fetchQuest.setExpReward(200);
fetchQuest.setGoldReward(100);

merchantComponent.addQuest(fetchQuest);
```

Step 3: Handle Quest Completion

The quest system will automatically:

- Show "!" when quest is available
- Show "..." when quest is in progress
- Show "?" when quest is complete

BUT you need to **restore GameLogic quest indicator management** for non-Fionne NPCs:

```java
// In GameLogic.update(), add this back ONLY for non-Fionne NPCs:
else if (entityType == EntityType.NPC) {
    NPC npcComponent = entity.getComponent(NPC.class);

    // ★ Skip Fionne - managed by IntroQuestHandler
    if (npcComponent != null && !"fionne".equals(npcComponent.getNpcId())) {
        updateNPC(entity, player, delta);
    }
}
```

Then restore the `updateNPC()` method for other NPCs.

Quick Reference

IntroQuestHandler Methods You Can Call

Method	Purpose
`forceSetStage(stage)`	Jump to a specific stage (debug)
`getCurrentStage()`	Get current quest stage
`isStageActive(stage)`	Check if on specific stage
`hasCompletedStage(stage)`	Check if past a stage
`resetIntroQuests()`	Start over (debug)

Where to Hook Quest Events

Event	Location	Method
Monster killed	`GameLogic.handleMonsterDeath()`	Call `introHandler.onMonsterKilled(type)`
Item collected	`UIManager.addItemToInventory()`	Call `introHandler.onItemCollected(item)`
Feature unlocked	`UIManager.unlockMenuButton()`	Call `introHandler.onFeatureUnlocked(id)`
Level reached	`GameLogic.awardExperience()`	Call `introHandler.onLevelReached(level)`

Best Practices

1. **Use IntroQuestHandler for**: Tutorial, feature unlocks, linear progression
2. **Use NPC Quest System for**: Side quests, repeatable quests, optional content
3. **Don't mix**: Keep Fionne's quests in IntroQuestHandler only
4. **Always update indicator**: Call `updateQuestIndicator()` after stage changes
5. **Test with debug keys**: Use N/M keys to test quest flow

Template for New Sequential Quest

```java
```

```
// 1. Add enum
STAGE_X_DIALOGUE,
STAGE_X_IN_PROGRESS,
STAGE_X_COMPLETE,

// 2. Add case to handleFionneInteraction()
case STAGE_X_DIALOGUE:
    showStageXDialogue(ui);
    return true;

// 3. Create methods
private void showStageXDialogue(UIManager ui) {
    // Show quest offer
}

private void completeStageX(UIManager ui) {
    // Grant rewards, advance stage
    currentStage = IntroStage.STAGE_X_COMPLETE;
    updateQuestIndicator();
}

// 4. Add tracking (if needed)
private int stageXProgress = 0;

public void onStageXEvent() {
    if (currentStage == IntroStage.STAGE_X_IN_PROGRESS) {
        stageXProgress++;
        // Check completion
    }
}

// 5. Update indicator switch
case STAGE_X_IN_PROGRESS:
    indicator.show(IndicatorType.IN_PROGRESS);
    break;
```

This gives you a complete framework for expanding quests!