Integration Guide: IntroQuestHandler

Step 1: Add to GameState.java

```java
package dev.main.state;

import dev.main.quest.IntroQuestHandler;

public class GameState {
    // ... existing fields ...

    private IntroQuestHandler introQuestHandler;

    public GameState() {
        // ... existing initialization ...

        // Initialize intro quest handler
        this.introQuestHandler = new IntroQuestHandler(this);
    }

    public IntroQuestHandler getIntroQuestHandler() {
        return introQuestHandler;
    }
}
```

Step 2: Update Engine.java - Remove Fionne-specific code

REMOVE these methods from Engine.java:

- `handleNPCClick()` - Replace with cleaner version below

REPLACE the handleNPCClick() method:

```java
```

```
private void handleNPCClick(Entity npc) {
    NPC npcComponent = npc.getComponent(NPC.class);
    if (npcComponent == null) return;

    Entity player = gameState.getPlayer();

    // Check range
    if (!npcComponent.isPlayerInRange(player, npc)) {
        System.out.println("Too far away to talk to " + npcComponent.getNpcName());
        return;
    }

    // ★ Check if intro quest handler wants to handle this interaction
    IntroQuestHandler introHandler = gameState.getIntroQuestHandler();
    if (introHandler != null && introHandler.handleFionneInteraction(npc)) {
        return; // Intro quest handled it
    }

    // ★ ORIGINAL CODE FOR OTHER NPCs (quest completion, active quests, dialogue)
    Quest completedQuest = npcComponent.getCompletedQuest();
    if (completedQuest != null) {
        gameState.getUIManager().showQuestComplete(npcComponent.getNpcName(), completedQuest);
        return;
    }

    Quest activeQuest = npcComponent.getActiveQuest();
    if (activeQuest != null) {
        gameState.getUIManager().showDialogue(npcComponent.getNpcName(), npcComponent.getDialogue());
        return;
    }

    // Otherwise fall back to dialogue database or greeting
    DialogueDatabase db = DialogueDatabase.getInstance();
    DialogueTree dialogue = db.getDialogueForNPC(npcComponent.getNpcId());

    if (dialogue != null) {
        UIDialogueBoxEnhanced dialogueBox = gameState.getUIManager().getEnhancedDialogueBox();
        dialogueBox.startDialogue(dialogue.getId(), npc, player);
    } else {
        gameState.getUIManager().showDialogue(
            npcComponent.getNpcName(),
            npcComponent.getGreetingDialogue()
        );
```

```
    }
}
```

Step 3: Update UIManager.java - Remove Fionne-specific code

REMOVE these methods from UIManager.java:

```java
// DELETE:
private boolean introQuestCompleted = false;
private boolean swordEquipped = false;
private boolean fionneSecondQuestAvailable = false;
private String fionneNotification = null;
private boolean secondQuestCompleted = false;

public boolean isFionneSecondQuestAvailable() { ... }
public boolean isIntroQuestCompleted() { ... }
public boolean hasSwordButNotEquipped() { ... }
public void showIntroDialogue() { ... }
public void showSecondQuestDialogue() { ... }
public void showEquipSwordReminder() { ... }
```

UPDATE equipItem() method:

```java
```

```
public boolean equipItem(UIGearSlot.SlotType slotType, Item item) {
    UIGearSlot slot = getGearSlot(slotType);
    if (slot == null) return false;

    Item oldItem = slot.equipItem(item);
    if (oldItem != null) {
        addItemToInventory(oldItem);
        applyItemStats(oldItem, false);
    }

    if (item != null) {
        applyItemStats(item, true);

        // ★ NOTIFY INTRO QUEST HANDLER WHEN WEAPON EQUIPPED
        if (slotType == UIGearSlot.SlotType.WEAPON && "Wooden Short Sword".equals(item.getName())) {
            IntroQuestHandler introHandler = gameState.getIntroQuestHandler();
            if (introHandler != null) {
                introHandler.onSwordEquipped();
            }
        }
    }
    return true;
}
```

Step 4: Add Debug Commands to Engine.java (Optional)

Add these to your `keyPressed()` method for testing:

```
java
```

```
// Debug: Advance intro quest (press N for "Next stage")
if (e.getKeyCode() == KeyEvent.VK_N) {
    IntroQuestHandler ih = gameState.getIntroQuestHandler();
    if (ih != null) {
        IntroQuestHandler.IntroStage current = ih.getCurrentStage();
        IntroQuestHandler.IntroStage[] stages = IntroQuestHandler.IntroStage.values();

        int nextIndex = (current.ordinal() + 1) % stages.length;
        ih.forceSetStage(stages[nextIndex]);

        System.out.println("DEBUG: Advanced intro quest to " + stages[nextIndex]);
    }
}

// Debug: Reset intro quest (press M for "reset Main quest")
if (e.getKeyCode() == KeyEvent.VK_M) {
    IntroQuestHandler ih = gameState.getIntroQuestHandler();
    if (ih != null) {
        ih.resetIntroQuests();
    }
}
```

Benefits of This Refactor

✅ Clean Separation

- All intro quest logic is in ONE place
- No scattered flags across multiple classes
- Engine/UIManager are now cleaner

✅ Easy to Extend

- Add new stages by following the template
- Each stage has clear: dialogue → completion → rewards
- Sequential progression automatically enforced

✅ Better State Management

- Single enum tracks ALL stages

- Helper methods check progression (hasCompletedStage, etc.)
- Debug tools to test each stage

✅ **Maintainable**

- Comments explain how to add new stages
- Template methods show the pattern
- No more hunting through files to find quest logic

How to Add Stage 4: "Kill 5 Slimes"

1. **Add to enum in IntroQuestHandler.java:**

```
java

STAGE_3_COMPLETE,
STAGE_4_DIALOGUE,     // ← Add this
STAGE_4_COMPLETE,     // ← Add this
ALL_COMPLETE
```

2. **Add case to handleFionneInteraction():**

```
java

case STAGE_3_COMPLETE:
    showStage4Dialogue(ui);
    return true;

case STAGE_4_COMPLETE:
    // Next stage or generic dialogue
    ui.showDialogue(npc.getNpcName(), "Well done, adventurer!");
    return true;
```

3. **Implement dialogue:**

```
java
```

```
private void showStage4Dialogue(UIManager ui) {
    UIDialogueBox dialogueBox = ui.getDialogueBox();

    dialogueBox.showMessageWithAccept(
        "Now, prove your strength. Slay 5 slimes that roam these lands.",
        () -> {
            // Create and add quest to quest log
            Quest slimeQuest = QuestFactory.createSlimeHuntQuest();

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

            currentStage = IntroStage.STAGE_4_DIALOGUE;
            dialogueBox.setVisible(false);
        },
        () -> dialogueBox.setVisible(false)
    );
}
```

4. **Complete when quest done (in quest completion code):**

```
java

private void completeStage4(UIManager ui) {
    // Grant rewards
    ui.addItemToInventory(ItemManager.createHealthPotion());
    ui.addItemToInventory(ItemManager.createHealthPotion());
    ui.addItemToInventory(ItemManager.createHealthPotion());

    // Unlock feature
    ui.unlockMenuButton("crafting");

    currentStage = IntroStage.STAGE_4_COMPLETE;
    System.out.println("[INTRO QUEST] Stage 4 Complete: Slimes defeated!");
}
```

Testing Checklist

- [ ] Talk to Fionne → Stage 1 dialogue appears
- [ ] Accept → Inventory unlocks with NEW badge
- [ ] Wooden sword appears in inventory
- [ ] Talk to Fionne again → "Equip the sword" reminder
- [ ] Equip sword → Stats button unlocks
- [ ] Talk to Fionne again → Stage 3 dialogue
- [ ] Accept → Rune received, blessing granted
- [ ] Talk to Fionne again → Generic "journey safe" dialogue
- [ ] Press N → Debug advance to next stage
- [ ] Press M → Debug reset to beginning

Migration Summary

Before:

- `introQuestCompleted` in UIManager
- `swordEquipped` in UIManager
- `fionneSecondQuestAvailable` in UIManager
- `showIntroDialogue()` in UIManager
- `showSecondQuestDialogue()` in UIManager
- Special case logic in Engine's `handleNPCClick()`

After:

- ✨ `IntroQuestHandler` with clear stage enum
- ✨ All logic centralized in one class
- ✨ Easy to add new stages
- ✨ Clean progression gates
- ✨ Debug tools included