# ADVANCED QUEST SYSTEM GUIDE

****Please note that this is only pseudocode – please do not try and run this as actual code in Godot!*

## 1. Setup Godot Project:

Make sure you have Godot installed. Create a new project or use an existing one where you want to add the quest system.

## 2. Create Quest Structure:

Define what data structure you'll use to represent quests. It's usually good to have a script that defines a Quest class. You can define this as a resource, which is useful for creating data that can be edited from the Godot editor.

```gd
# Quest.gd

extends Resource

class_name Quest



export(String) var title

export(String) var description

export(int) var xp_reward

export(bool) var is_complete = false

# You can add more properties such as objectives, sub-quests, etc.
```

Don't forget to save this script as a resource by clicking the save icon next to the script name in the script editor.

### 3. **Create Quest Manager:**

This will be an autoload singleton that keeps track of active and completed quests.

```gd
# QuestManager.gd

extends Node

class_name QuestManager



var active_quests : Array = []

var completed_quests : Array = []



func add_quest(quest: Quest):

    active_quests.append(quest)



func complete_quest(quest: Quest):

    quest.is_complete = true

    active_quests.erase(quest)

    completed_quests.append(quest)

# Add functions to query for quest status, etc.```
```

4. **Setup Autoload Singleton:**

Go to Project > Project Settings > Autoload. Add the `QuestManager.gd` script and make sure it is enabled. This way, you can access the Quest Manager from any script.

5. **Create Quest Instances:**

You can create quest instances from the Godot editor by creating new Resources using your Quest script, or you can create them in code. To create them in code, you can do something like this:

```gd
var new_quest = Quest.new()
new_quest.title = "Find the Key"
new_quest.description = "Find the key in the forest."
new_quest.xp_reward = 100
QuestManager.add_quest(new_quest)
```

6. **Update and Display Quests:**

Your game logic should update quests' status and objectives as the player makes progress. You also need a GUI to display active quests to the player. Use `Control` nodes (like Labels and Panels) to build a UI that shows the player's quests. You can access the quest data via the QuestManager singleton.

7. **Saving and Loading:**

Implement saving and loading the quest data to keep track across game sessions. You can use Godot's built-in `FileAccess` class or the `JSON` class for saving and loading quest data.

This is a basic outline of how to set up a quest system in Godot. Depending on your game, you may need to add more features such as branching quests, quest dependencies, NPCs giving quests, etc. The principles will remain the same; create data structures to represent the quests, manage them efficiently, and display information to the player in an engaging way.