

# Adventure Game Project – Expanded Redesign Guide

## 1. Object-Oriented Structure

Create a base class ``Scene`` with a method ``enter()``.

Each room (e.g., ``SpiderRoom``, ``PythonRoom``, ``ChestRoom``) inherits from ``Scene`` and overrides ``enter()``.

This gives a clean, reusable structure and makes it easy to add/remove rooms.

## 2. Player State (No Globals)

Replace the ``key`` global variable with a ``Player`` object that has attributes like:

- inventory (list of items collected)
- health (numeric value for survival)
- status (flags like 'has\_key', 'is\_wounded').

Pass the ``Player`` object into ``enter()`` so each scene can modify it.

## 3. Engine as the Core

The Engine runs the game loop:

- Calls the current scene's ``enter()``.
- Waits for user input.
- Gets the name of the next scene.
- Loads the next scene from the ``Map``.

Stops when it reaches the ``finished`` scene.

## 4. Modular Project Layout

engine.py – main game loop.

map.py – dictionary mapping scene names to scene objects.

scenes.py – all room classes.

player.py – player state.

utils.py – helper functions (e.g., formatted text, math puzzles).

Optional: data/ folder with descriptions or puzzles.

## 5. Using Built-in Modules Effectively

random – pick outcomes (e.g., safe pod, monster reaction).

math – integrate puzzles (e.g., unlock a door by solving `sqrt(144)`).

sys – use `sys.exit()` for quitting, or `argv` for starting parameters.

textwrap – format long text descriptions for readability.

## 6. Extensibility

Add new scenes just by writing a new Scene subclass and updating the Map.  
Swap descriptions from Python code into external files later (e.g., JSON or text).  
Add difficulty modes (easy/hard) by controlling randomness or puzzles.

## 7. Suggested Development Roadmap

1. Convert your existing functions (spider\_room, python\_room, etc.) into Scene subclasses.
2. Build a simple Player class with just an inventory.
3. Set up engine.py with the loop from stage 1.
4. Split files into map.py, scenes.py, etc.
5. Gradually replace hardcoded logic with helpers from utils.py.
6. Expand gameplay (health, multiple items, more puzzles).