Modular Structure

The following document shows different modules of the game, as a base for the identification of classes or methods during programming.
Note: the diagram below is NOT a class diagram. It rather shows the structure of different components of the game, datatypes that might be used and how these components will interact with each other.

Room
- Level, in which the player interacts with characters in the story or items
- Levels will have background images, that show the scenery
  - ➢ Items that can be interacted with will be marked by a number
  - ➢ By pressing the equivalent number on the keyboard, the user selects the item he wants to interact with

Player
- Character the user plays, gives the user a place in the gaming world

WorldObject
- Item in the gaming world that the player can add to the inventory or combine with other items of his/her inventory

InventoryObject
- Item in the inventory of the player that can be combined with other items of the inventory or items of the gaming world
- By combination with other items, new items can be made

Inventory
- Collection of all items that the player finds or creates in the gaming world

EventHandler
- monitors which numbers are pressed on the keyboard
- Classes/methods in the code can subscribe to the EventHandler
  - ➢ corresponding actions are triggered in the subscribing classes/methods

Music
- Background music
- GEMA-free
- to emphasize the atmosphere of the game

Music
title: mp3

EventHandler
userInput: int

Room
name: string
worldObjects: list[WorldObject]
text: string
interaction: userInput
backgroundImage: string (ASCII)
music: mp3

WorldObject
name: string
useWith: string

InventoryObject
name: string
useWith: string
resultingObject: InventoryObject

Inventory
Items: list[InventoryObject]

Player
name: string
backgroundStory: string
interaction: userInput
location: RoomObject

contains
subscribe
subscribe
subscribe
subscribe
subscribe
contains
used with
is in
interacts with
interacts with
add
add
interacts with