

Text Adventure game in Python C# and Lua

Create Game Objects

These all have multiple instances: use classes with constructors

1. Create Items first: Items do not contain any objects (bottom of the chain)

Weapons, keys, and other objects that can be carried or used by the player and can be found in locations, or dropped from enemies when they are defeated.

Items have these properties:

name, description, damage (the damage they cause if used as a tool/weapon)

2. Create enemies next: enemies can contain Items (need items defined already)

Enemies have these properties:

name, description, health, strength, drop item (when defeated)

Enemies have these methods:

attack

3. Create locations (geographical areas, cells, rooms etc.)

Can contain both items and enemies, so these need to be created first

Locations have these properties:

name, display name, description, location to n/e/s/w, item(s), enemy(s)
item required to enter the location (key etc.)

Locations have these methods:

add/remove an item

add/remove an enemy

display a description of the location and its exits / items

Set the surrounding locations

4. Create Player

Only one Instance: use a static class or module

Player has these properties:

name, character, age group, strength, health, inventory, item in hand

Player methods:

add/remove <--> item <--> hand/inventory

5. Create shared area for global variables and shared methods

Only one Instance: use a static class or module

Player has these properties:

name, character, age group, strength, health, inventory, item in hand

Player methods:

add/remove <--> item <--> hand/inventory

6. Create Game Setup

Only one Instance: use a static class or module

Player has these properties:

name, character, age group, strength, health, inventory, item in hand

Player methods:

add/remove <--> item <--> hand/inventory

7. Create Console IO class

Only one Instance: use a static class or module

For creating the player and navigating the game, the user(player) will need to respond to requests for input. To ensure there are no errors, either accidental or deliberate, a robust keyboard input class is required. Suggested name 'kboard' (Input is a Python keyword so not suitable)

Methods of kboard class:

Return string (with option of Title case), integer, float, boolean values

Display a menu system and return the index of the chosen item