

Dungeon Adventure

The Spoony Bard

Team: Halt Catch Fire

12/11/2021

Adventurer
<div>- name : str</div> <div>- dev_powers : bool</div> <div>- max_hit_points : int</div> <div>- current_hit_points : int- health_pots : int</div> <div>- vision_pots : int</div> <div>- pillars_collected : {str : bool}</div>
<div>+ __init__(self, name : str, challenge : str)</div> <div>+ is_alive(self) : bool+ has_all_pillars(self) : bool</div> <div>+ add_potions(self, room_potions : tuple)</div> <div>+ add_pillar(self, pillar : str)</div> <div>+ damage_adventurer(self, pit_damage : int)</div> <div>+ heal_adventurer(self, heal_amount : int)</div> <div>+ use_health_potion(self)</div> <div>+ use_vision_potion(self)</div> <div>- _create_adventurer(self, name : str, challenge : str)</div> <div>- _readable_pillars(self)</div> <div>+ __str__(self)</div> <div>+ __repr__(self)</div> <div>+ property(name, name.setter)</div> <div>+ property(dev_powers, dev_powers.setter)</div> <div>+ property(max_hitpoints, max_hitpoints.setter)</div> <div>+ property(current_hitpoints, current_hitpoints.setter)</div> <div>+ property(health_pots, health_pots.setter)</div> <div>+ property(vision_pots, vision_pots.setter)</div> <div>+ property(pillars_collected)</div>

Health Pots(Potions)
<div>- name : str</div> <div>- heal_amount : int</div>
<div>+ __init__(self, random : bool)</div> <div>+ __str__(self)</div> <div>+ __repr__(self)</div> <div>- _potion_effect(self)</div> <div>+ action(self)+ property(name, name.setter)</div> <div>+ property(heal_amount, heal_amount.setter)</div>

Potion(abc)
<div>+ name(self)</div>
<div>+ __init__(self)</div> <div>+ __str__(self)</div> <div>+ action(self)</div> <div>- _potion_effect(self)</div> <div>+ property(name)</div>

PotionFactory
<div>+ create_potion(name : str) : Potion</div>

Health Pots(Potions)
<div>- name : str</div> <div>- rooms_revelaed : int</div>
<div>+ __init__(self, random : bool)</div> <div>+ __str__(self)</div> <div>- _potion_effect(self)</div> <div>+ action(self)+ property(name, name.setter)</div> <div>+ property(rooms_revealed, rooms_revealed.setter)</div>

Main
<div>+ __init__(self)</div> <div>+ game_flow(self)</div> <div>+ print_game_menu(self)</div>

Map
<div>+ visited : [] : boolean</div> <div>+ __str__(self)</div>
<div>+ method(type): type</div>

Room
<div>- health_potion : int</div> <div>- vision_potion: int</div> <div>- doors : dict</div> <div>- pit : int</div> <div>- contents : str</div>
<div>+ __str__(self) : bool</div> <div>+ __eq__(self) : bool</div> <div>- __is_number_gt_eq_0(num) : bool</div> <div>- __is_boolean(boolean : bool) : bool</div> <div>- __is_valid_contents(contents : str) : bool</div> <div>- __is_valid_creation_data() : bool</div> <div>- __update_room_content(self) : bool</div> <div>+ string_top(self) : str</div> <div>+ string_middle(self) : str</div> <div>+ string_bottom(self) : str</div> <div>+ can_enter(self) : bool</div> <div>+ clear_room(self)</div> <div>+ property(exit, exit.setter)</div> <div>+ property(entrance, entrance.setter)</div> <div>+ property(health_potion, health_potion.setter)</div> <div>+ property(vision_potion, vision_potion.setter)</div> <div>+ property(pit_damage)</div> <div>+ get_door(self, direction) : bool</div> <div>+ set_door(self, direction, door_exists) : bool</div> <div>+ property(contents, contents.setter)</div> <div>+ property(visited)</div>

DungeonBuilder
<div>- rowCount : int</div> <div>- colCount : int</div>
<div>- __reset(self, difficulty : str, varied : bool)</div> <div>- __set_dungeon(self)- __get_rand_coords(self) : tuple</div> <div>- __build_2d_room_maze(self)</div> <div>- __build_room(self) : Room</div> <div>- __build_dungeon_path(self, row : int, col : int)</div> <div>- __build_pillars(self)</div> <div>- __is_traversable(self, row : int, col : int)</div> <div>- __is_valid_room(self, row : int, col : int)</div> <div>- __get_room(self, coordinates) : Room</div> <div>+ build_dungeon(self, difficulty : str, varied : bool) : Dungeon</div> <div>+ __init__(self, difficulty : str, varied : bool)</div>

Dungeon(Iterable)
<div>- rowCount : int</div> <div>- colCount : int</div>
<div>+ __init__(self, dungeon : list, difficult : str, ent : tuple, ex : tuple)</div> <div>+ __str__(self)</div> <div>+ __iter__(self)</div> <div>+ __eq__(self)</div> <div>- __is_valid_room(self, row : int, col : int)</div> <div>+ get_room(self, coordinates)</div> <div>+ property(total_rows) : int</div> <div>+ property(total_columns) : int</div> <div>+ property(entrance) : tuple</div> <div>+ property(exit) : tuple</div> <div>+ property(pillars) : str</div> <div>+ property(dungeon) : [ [] ]</div> <div>+ property(adventurer_loc) : tuple</div> <div>+ property(pit_damage) : int</div> <div>+ move_adventurer(self, direction : str)</div> <div>+ collect_potions(self) : tuple</div> <div>+ collect_pillars(self) : str</div> <div>+ get_visible_dungeon_string(self, bool_list : [ [] ]) : str</div>

DungeonIterator(Iterator)
<div>- collection : list</div> <div>- row : int</div> <div>- col : int</div> <div>- col_count : int</div>
<div>+ __init__(self, dungeon : list, col_count : list)</div> <div>+ __next__(self)</div>

Steph:

Room

Dungeon

DungeonBuilder

Kevin:

Adventurer

Potion

Potion Factory

Health Potion

Vision Potion

Xingguo:

Main

Map

