Dungeon Adventure The Spoony Bard Team: Halt Catch Fire 12/11/2021

### Map

- + visited : [][] : boolean
- + \_\_str\_\_(self)
- + method(type): type

#### Main

- + \_\_init\_\_(self)
- + game\_flow(self)
- + print game menu(self)

# Adventurer

Steph:

Room

Kevin:

Main Мар

Dungeon

Adventurer Potion

Potion Factory

Health Potion

Vision Potion Xingguo:

DungeonBuilder

- name : str
- dev powers : bool
- max hit points : int
- current\_hit\_points : int- health\_pots : int
- vision\_pots: int
- pillars\_collected : {str : bool}
- + \_\_init\_\_(self, name : str, challenge : str)
- + is\_alive(self) : bool+ has\_all\_pillars(self) : bool
- + add potions(self, room potions : tuple)
- + add pillar(self, pillar : str)
- + damage adventurer(self, pit damage : int)
- + heal\_adventurer(self, heal\_amount : int)
- + use\_health\_potion(self)
- + use vision potion(self)
- create adventurer(self, name : str, challenge : str)
- readable pillars(self)
- + \_\_str\_\_(self)
- + \_\_repr\_\_(self)
- + property(name, name.setter)
- + property(dev\_powers, dev\_powers.setter)
- + property(max hitpoints, max hitpoints.setter)
- + property(current\_hitpoints, current\_hitpoints.setter)
- + property(health\_pots, health\_pots.setter)
- + property(vision\_pots, vision\_pots.setter)
- + property(pillars\_collected)

# DungeonBuilder

- rowCount : int - colCount : int
- + method(type): type

# PotionFactory

reate potion(name : str) : Potion

Potion(abc)

+ \_\_str\_\_(self)+ action(self)

+ name(self)

+ \_\_init\_\_(self)

+ property(name)

- \_potion\_effect(self)

## Health Pots(Potions)

name : str

- heal amount : int

- + \_\_init\_\_(self, random : bool)
- + \_\_str\_\_(self)
- + \_\_repr\_\_(self)
- potion effect(self)
- + action(self)+ property(name, name.setter) + property(heal\_amount, heal\_amount.setter)

# **Health Pots(Potions)**

- name : str
- rooms revelaed : int
- + \_\_init\_\_(self, random : bool)
- + \_\_str\_\_(self)
- potion effect(self)
- + action(self)+ property(name, name.setter)
- + property(rooms revealed, rooms revealed.setter

## Room

- pit : int

- is boolean(boolean : bool) : bool
- \_\_is\_valid\_contents(contents:str):bool

- + string middle(self) : str
- + can enter(self) : bool
- + clear\_room(self)
- + property(exit, exit.setter)
- + property(entrance, entrance.setter)
- + property(health potion, health potion.setter)
- + property(vision potion, vision potion.setter)
- + get\_door(self, direction) : bool
- + set\_door(self, direction, door\_exists) : bool
- + property(contents, contents.setter)
- + property(visited)

- health\_potion: int
- · vision\_potion: int
- doors : dict
- contents : str
- + \_\_str\_\_(self) : bool
- + \_\_eq\_\_(self) : bool
- \_\_is\_number\_gt\_eq\_0(num) : bool
- is\_valid\_creation\_data() : bool
- \_\_update\_room\_content(self) : bool
- + string top(self): str
- + string bottom(self) : str

- + property(pit\_damage)

## Dungeon(Iterable)

- rowCount : int colCount: int
- + method(type): type

# **DungeonIterator(Iterator)**

- rowCount : int - colCount : int
- + method(type): type