HSMA 4 Python Assessment – "What a Knightmare!"

GENERAL INSTRUCTIONS

From 1987 – 1994, ITV broadcast a children's TV show that became a cult classic. The show was called Knightmare. A team of children played the game, in which one of the team became the "Dungeoneer", putting on the "Helmet of Justice" (which blocked their view) and they entered the "dungeon". Their teammates watched on from a monitor, giving directions and instructions to the dungeoneer, telling them where to walk, what to do etc. To the teammates, it appeared the dungeoneer was adventuring through an elaborate series of rooms in a dungeon via live graphics (when in actual fact, they were just walking around a rather bland series of mostly empty rooms).

In their adventure, they'd often be presented with various items that they could choose to pick up and put in their adventuring pouch. They'd also meet a cast of characters that would give them riddles to solve, as well as ask them for items. Usually an item that they had to leave behind some time ago, and had no way of knowing they'd actually need. Very few teams ever got to the end…

Your task is to write a simple program that emulates some of the aspects of the show Knightmare. You should follow the instructions below carefully. You may refer to any of your notes from the course during the training (you only need to refer to the slides from sessions 1C, 1D and 1E), but **you must work alone –** plagiarism will result in disqualification from the assessment.

You should create a new .py file, and name it with your name (e.g. dan_chalk.py). Once you have completed the assessment, you must send me the .py file as a private message on Slack. To do this, simply click the white pen and paper icon at the top left of Slack, type my name (until it appears, then click it) then drag and drop your .py file into the message area, and send the message.

You have 1 hour to undertake the assessment. You may finish early. You **must** submit a .py file to me no later than 5 minutes from the end of the assessment. Failure to do so will result in disqualification. So please send something even if it is incomplete.

You will not be formally marked for this assessment. If your project pitch is shortlisted for selection, your assessment will be looked at just to check that the code you wrote gives enough indication that you have sufficiently picked up the basics of Python coding to undertake your project. **You do not need to undertake the assessment if you are not submitting a project proposal as the Project Manager. This also means that trainee mentors do not need to take the assessment.**

THE ASSESSMENT

1. Import the random library, as well as numpy (using the conventional alias)

2. Create a class called *Dungeon*, which contains the following :

- A constructor that accepts a passed in dungeon name (don't forget to pass in self too!), and which sets up two attributes for this class. The first a 2D numpy array containing the numbers 1 – 9 in three groups (ie [1,2,3], [4,5,6] etc) named *dungeon_map*. The second a string variable called *dungeon_name*, which is set to the passed in dungeon name value. Once both attributes have been set up, the constructor should print a message saying that the dungeon has been created, and should include the name of the dungeon in the printed message.

3. Create a class called *Room*, which contains the following :

- A constructor that accepts a passed in room number, dungeon object (which specifies in which dungeon the room resides), and list of items (as strings). The constructor should set up three attributes for the Room class named *room_num, dungeon* and *item_list*, and store the corresponding values passed in

- A method named *display_available_items* which prints a message saying something along the lines of "here's a list of items in this room", before printing each item stored in the item list for this room in turn, using a loop.

4. Create a class called *Dungeoneer*, which contains the following :

- A constructor that accepts a passed in name for the dungeoneer. The constructor should set up five attributes for the Dungeoneer class – *name* (which will store the dungeoneer name passed in), a Boolean *helmet_of_justice_on* (which should default to False), a list called *pouch* (which represents the items the dungeoneer is carrying, and which should default to being empty), an integer named *current_room* (which represents the room number the dungeoneer is in, and which should default to 0), and a string named *current_dungeon_name* (which represents the name of the dungeon the dungeoneer is in, and which should default to an empty string).

- A method named *put_on_helmet_of_justice*, which should switch the Boolean *helmet_of_justice_on* attribute to True

- A method named *enter_dungeon*, which should accept a dungeon object as an input. The method should check whether the dungeoneer's helmet of justice is on. If it is, then it should print a message saying they are entering the dungeon (and give the name of the dungeon in the printed message), update the dungeoneer's current dungeon name accordingly, and return a Boolean value of True. If the dungeoneer's helmet is not on, then a message should be printed saying that the dungeoneer cannot enter the dungeon without the helment of justice, and should return a Boolean value of False.

- A method named *pick_up_item*, which should accept a room object as an input. The method should a) use the appropriate method of the room object to display the items in the room, b) choose an item at random from the items in this room, c) print a message saying which item has been taken, d) remove the item that has been taken from the available items in the room, e) add the item to the dungeoneer's pouch.

5. Once you have written the class definitions above, you should write code at the end (after the definitions) that does the following :

- Creates a new dungeon object, and gives it a name of your choosing
- Creates a new room object, that is room number 8, resides in the dungeon you created above, and contains four items : bread, an apple, a potion, and a scroll.
- Creates a new dungeoneer with your name
- Gets the dungeoneer you created to put on their helmet of justice
- Gets the dungeoneer you created to enter the dungeon you created
- Checks whether the dungeoneer successfully entered the dungeon (remember – calling the method to enter the dungeon will return a value…). If they did, then get the dungeoneer to pick up two items, one after the other. Then print the contents of their pouch. If they didn't successfully enter the dungeon, do nothing.