Tips for programming IA2

This are some of my thoughts about how I programmed my UltraSuperMegaTrumps game. My program only incudes the card game, and no details about the user.

Create Virtual Environment

Make sure that you create a virtual environment, just like in our Hangman exercises.

Class structure

I used the following classes:

- MainWindow class - the control module
- SuperheroDB class - the model (datastore) module for the Superheroes
- Ui_MainWindow class - the view module (generated from the QT Designer file)
- Card class - a class to store the information of each card

Development Order

I developed my program in the following way

1. The SuperheroDB class:
    - build the database structure
    - import the values from the CSV file
    - test using a testing.py module which outputted to the terminal
    - create insert methods to add data when necessary
    - create get methods for queries when necessary
2. The Card class
    - create method to add card stats
    - create method to return card stats
    - test using a testing.py module which outputted to the terminal
    - create SuperheroDB get methods for queriers when necessary
3. The Ui_MainWindow class
    - generate from the QT Designer file
    - do not change anything
    - regenerate every time the QT Designer file changes
4. The MainWindow class
    - use the PyQt 6 Boilerplate code in Teams (MainWindow.py)
    - this is were the the game logic occurs.
    - create SuperheroDB and Card methods as needed

Using Lists

List are the perfect data structure to represent decks of cards.

For this project I used a list containing card objects to represent:

- The deck (pack)
- Player's hand (player_hand)

- Computer's hand (ai_hand)

Use list methods and operations to work with your various lists.

For example, to deal the hands

```
while len(deck) > 0:
    dealt_card = deck.pop()
    player_hand.append(dealt_card)
    dealt_card = deck.pop()
    ai_hand.append(dealt_card)
```

AI Logic

The logic I chose to use for the AI was:

- Go through the playing deck (when selected) and, for each card, rank each of it's statistics.
- Statistics are stored in order in a ranking list
- When the AI chooses, it selects a statistic from the list:
    - Easy difficulty - randomly choose from all stats
    - Medium difficulty - randomly choose from the top three stats
    - Hard - always choose from the top stat

The pseudocode for ranking the cards:

```
BEGIN rank_card(card,deck)

SET all stat rankings to 1

FOR index = 1 TO len(deck)
    FOR each stat
        IF card[stat] < deck[index] stat
            INCREASE stat ranking by 1
        ENDIF
    NEXT stat
    ENDFOR
NEXT index
ENDFOR

SORT card stats according to ranking

RETURN card stats

END rank_card
```

Accessing the images

The images for the cards have been provided on Teams. They have been named after the file name in in the URL.

For example, Godzilla's image URL is
`https://www.superherodb.com/pictures2/portraits/10/100/10590.jpg`

The file for Godzilla is name `10590.jpg`

Manipulating values from the CSV

Values are read from the CSV as a `string`, therefore you will need to utilize string methods and operators to manipulate them to produce what you want.