About

Introduction

The methodology of School 21 makes sense only if peer-to-peer reviews are done seriously. Please read all guidelines carefully before starting the review.
- Please, stay courteous, polite, respectful and constructive in all communications during this review.
- Highlight possible malfunctions of the work done by the person and take the time to discuss and debate it.
- Keep in mind that sometimes there can be differences in interpretation of the tasks and the scope of features. Please, stay open-minded to the vision of the other.
- If you have not finished the project yet, it is compulsory to read the entire instruction before starting the review.

Guidelines

- Evaluate only the files that are in src folder on the GIT repository of the student or group.
- Ensure to start reviewing a group project only when the team is present in full.
- Use special flags in the checklist to report, for example, an “empty work” if repository does not contain the work of the student (or group) in the src folder of the develop branch, or “cheat” in case of cheating or if the student (or group) are unable to explain their work at any time during review as well as if one of the points below is not met. However, except for cheating cases, you are encouraged to continue reviewing the project to identify the problems that caused the situation in order to avoid them at the next review.
- Doublecheck that the GIT repository is the one corresponding to the student or the group.
- Meticulously check that nothing malicious has been used to mislead you.
- In controversial cases, remember that the checklist determines only the general order of the check. The final decision on project evaluation remains with the reviewer.

Main part

Exercise 00 - Talk and Fight

- Python files with the code are present

- `Protagonist`, `NPC` and `Enemy` classes are implemented
- Protagonist class skeleton is used where methods 'talk_to()', 'take_hit()', 'attack()', 'take()' and 'give()' are implemented
- It is not possible to attack an NPC or take an item from the undefeated enemy
- A battle mechanic is implemented such that a random number from 1 to 6 is generated and added to a current Protagonist's skill level. The same happens for the Enemy. If Protagonist wins, Enemy is defeated. If Enemy wins, Protagonist loses one or more HP
- Script 'load_data.py' is provided that allows to populate the database with NPC and Enemy stats, phrases and interactions
- Player profile is also initialized in the database (automatically or via user 'registration') that holds the current inventory and stats
- There is a simple way (maybe with some additional tutorial) to create an instance of a specific NPC from the database and talk/exchange items with this character
- There is a simple way (maybe with some additional tutorial) to create an instance of a specific Enemy from the database and initiate a fight, which leads to a winning message or HP decrease for the Protagonist

👎 0 1 2 3 4 5 👍

Exercise 01 - I Like to Move It Move It

- Python files with the code are present
- `Direction` type is implemented
- Script 'load_map.py' is provided that allows to populate the database with the graph of game locations
- Methods 'go()' and 'whereami()' are implemented for a Protagonist class
- It is impossible to go to a location which is not directly connected to the current one
- A location description is printed when Protagonist enters a new place

👎 0 1 2 3 4 5 👍

Exercise 02 - Don't Shoot the Messenger

- Python files with the code are present
- A Telegram bot is implemented using either 'telebot' or 'aiogram' Python package
- A set of buttons is shown to user when bot is activated
- A menu with subcategories has at least two levels with a button to go up a level
- Different methods/databases are used for generating names for different subcategories of characters and items
- After the name is generated and returned by a bot a user is back at the top menu level

- Creator's bot token is NOT included into the submission. If it is, please notify the author so he could remove the bot from his account and re-generate the token

👎 0 1 2 3 4 5 👍

Exercise 03 - The Whole Story

- The game Python files are present
- Script 'load_all.py' is provided that allows to populate the database with the game data
- Project includes a Sphinx-powered documentation that can be generated using "make html" as in DAY07
- Documentation includes sections on starting the bot and a game walkthrough
- It is actually possible finish the game using the walkthrough (though it is really recommended to play without it first, for fun)
- The game includes all the required working mechanics from previous exercises (all of them NOT necessarily present in a single playthrough, but on possible branches), like moving through the world, talking to NPCs, exchanging items with them, fighting enemies and advancing in skills
- Creator's bot token is NOT included into the submission. If it is, please notify the author so he could remove the bot from his account and re-generate the token

👎 0 1 2 3 4 5 👍

Bonus part

Exercise 00 - Bonus part

- Protagonist can get quests from NPCs, complete them and receive the reward

👎 0 1 2 3 4 5 👍

Exercise 01 - Bonus part

- Bot is using async/await for implementing asynchronous methods to react to user's interaction

👎 👍