

Task 2 for Professional Developers: Snake

Description

You are provided with a simple Python implementation of the classic Snake game. You can try running it by typing in the terminal `python3 main_game.py`. Your task is to add four features to the provided implementation:

1. **Compute score:** The current implementation does not keep track of the user's score. Implement the logic needed to increase the score for each piece of food eaten. The score starts at 0, and it is incremented by a +1 for each eaten piece of food. Every 10 eaten pieces, there is a further bonus of +5. The score must be shown on screen while the user is playing (bottom-right corner).
2. **Wrap-around behavior:** Currently, the player loses if the snake collides with one of the four walls. Remove such a behavior, and implement a wrap-around logic: If the snake "collides" with a wall, its head is moved to the opposite side, and the game can continue.
3. **Starting page:** Currently, the game immediately starts when `main_game.py` is lunched. Modify such a behavior so that when lunching `main_game.py` the user is asked for a non-empty nickname. Once entered, if the user presses return the game starts. If, instead, he presses "1", the top-10 all time scores are shown on screen (see next point).
4. **Storing top-10 all time scores:** When the user loses, its score is saved in a CSV file (`scores.csv`) assuming it is one of the top-10 scores of all time. The CSV features the following columns:
 - Username, score, timestamp.

Checklist for Evaluation

- ☐ Implements the logic needed for keeping track of the score. (15%)
- ☐ Score is correctly displayed on screen. (5%)
- ☐ Wrap-around behavior is implemented. (20%)
- ☐ User name is asked to the user at the start of the program. (15%)
- ☐ Loads and visualizes the top-10 scores when asked. (15%)
- ☐ Saves the top-10 scores to `scores.csv` when the player loses. (20%)
- ☐ Exception handling (e.g., file not found, empty user name). (10%)