Dungeon Version 3: Exploring

Time to move!

In this version, you will implement moving on the map (going north/south/west/east).

What to do

In addition to the functions you already have:

1. Implement the `move` function that would move the player to a specified direction. This function should call the `look_around` function to determine if the move is valid or not. If it is not valid, the function should return `False`, otherwise, the function should change the player position according to the move and return `True`.

Example of the displayed map:

```
**@--
*----
***--
--**F
```

Player's position: `[0, 2]`

Calling the function as follows would return `True` and change the player's position (the `player_position` list) to `[0, 1]`:

```
success = move('west', position, grid)
```

See the hint about this function!

2. Update the `main` function:

> In addition to the **escape** and **show map** commands, it should now accept the following four commands: **go north, go south, go west, go east**.
> Upon any of these commands, the player should attempt to move in a given direction. If the attempt is successful, the program should display **You moved <direction>.** Otherwise, it should display **There is no way there.**

Use the following template. All functions defined in the template **must be present and implemented** in your code (you may **not** omit functions or change the function definitions). You **may** add extra functions if needed.

```python
MAP_FILE = 'cave_map.txt'

def load_map(map_file: str) -> list[list[str]]:
    """
    Loads a map from a file as a grid (list of lists)
    """
    # Implemented in version 1

def find_start(grid: list[list[str]]) -> list[int, int]:
    """
    Finds the starting position of the player on the map.
    """
    # Implemented in version 1

def get_command() -> str:
    """
    Gets a command from the user.
    """
    # Implemented in version 1

def display_map(grid: list[list[str]], player_position: list[int, int]) -> None:
    """
    Displays the map.
    """
    # Implemented in version 2

def get_grid_size(grid: list[list[str]]) -> list[int, int]:
    """
    Returns the size of the grid.
    """
    # Implemented in version 2

def is_inside_grid(grid: list[list[str]], position: list[int, int]) -> bool:
    """
    Checks if a given position is valid (inside the grid).
    """
    # Implemented in version 2

def look_around(grid: list[list[str]], player_position: list[int, int]) -> list:
    """
    Returns the allowed directions.
    """
    # Implemented in version 2

def move(direction: str, player_position: list[int, int], grid: list[list[str]]) ->
bool:
    """
    Moves the player in the given direction.
    """
    # TODO: implement this function

def main():
    """
    Main entry point for the game.
    """
    # TODO: update the main() function

if __name__ == '__main__':
    main()
```

Hints

- Your `move` function should check if the player can move in the given direction and only then, actually make the move. You already have a function that does the check! Don't repeat yourself by writing all checks again, just call your `look_around` function in `move`!

Program name

Save your program as `dungeon3.py`.

Demo

In this demo, `cave_map.txt` is used.

https://asciinema.org/a/drARq55xIJ3LNAFDyjrM2I2hy

Testing

To make sure your program works correctly, you should test it.

Good news: we wrote the unit tests for you: **test_dungeon3.py**

To test your functions, simply run the unit tests:

```
$ python -m pytest test_dungeon3.py
```

All tests should pass.

Submitting

Submit `dungeon3.py` via eClass.