THE UNIVERSITY OF SYDNEY

INFO1110 / COMP9001 Acorn Runner

Amendments / Clarifications / Changelog

v1.3 -> v1.4

- Added file architecture diagram.
- Updated scaffold e2e script and folder.

v1.2 -> v1.3

- If parse() encounters Teleport pad '0', it should raise the "Bad letter" error.
- Clarified that on bad user input, the game should continue and ask for the next input.
- Added: `You are expected to create the other unit test files yourself. If you believe that a file/module is not worth unit testing, create the file and justify it as a comment why you believe you don't need to unit test it!` to the student testing section.
- Added: `Please only use the cell's display attribute to pass the test cases in parse().` to the Milestone section.
- The output on game finish should say `You made X moves.`, not `Your made X moves.`
- Updated BFS sample output to be the correct output.

v1.1 -> v1.2

- "If the file doesn't exist, print `does not exist!` and exit gracefully." Should be under the `read_lines()` spec. Not under `parse()`.
- The following paragraph should be under `game.py`, not `grid.py`:
    - You should call `read_lines()` which uses the `parse()` function to parse the lines in the file. Your `read_lines()` function should return the grid as well. If you decide to do this differently however, that is also ok!
- `game_parser.py`'s `parse()` function must be able to handle `'\n'` character at the end of each string in the given list.
- The Player class must have a `row` and `col` attribute which represents their location on the grid. The `grid_to_string()` function must use these attributes when drawing the player.
- If a Player tries to leave the grid (say there was a hole in the perimeter), it should act as if the player walked into a wall.
- Added the following section:
    - The order of function calls in this file should look like:

        - Call read_lines()
            - Call parse(lines)
            - Return grid
        - Return grid
- Scaffold changes:
    - Renamed `gameMove()` to `game_move()`.
    - Added skeleton code for `Player.row` and `Player.col`.

v1.0 -> v1.1

- If a file does not exist, print `does not exist!` and exit gracefully.
- If you're on a teleport pad and you walk into a wall, the wall pushing you back does not re-trigger the teleport. (Remember, moves into walls should not be recorded!)
- The `parse()` function in `game_parser.py` only needs to output a single unknown letter, if encountered. You do not need to find them all and concatenate to a single message. In reality when your program is used, if a user can fix it up and see the next error message, that would be sufficient :). If you want to parse the whole thing and generate a single string containing all the error messages as an extension, feel free to!
- Similar to above, the `parse()` function only needs to output one pad number that does not have an exclusively matching pad.
- FAQ: You may use `for` and `in` in this assignment!
- "Walking into a wall" sample output should say 2 moves made, with `d, d` as the moves.
- The `read_lines()` function in `game_parser.py()` should call `parse()` and return the grid. However it is also ok if you decide to do this differently!
- Scaffold changes:
    - Changed doc string in read_lines() to now say: """Read in a file, process them using parse(), and return the contents as a list of list of cells."""