Functional Specifications:

The program is an adventure game that starts in a jungle environment. The player can navigate through the jungle by choosing directions (north, south, east, west) or quit the game. The player's goal is to explore the jungle and find new environments The game keeps track of the number of times the player travels in each direction to trigger discovery of environments.

### this sucks

The game introduces a mini-game feature when the player decides to travel in a specific direction. The mini-game varies depending on the environment (jungle or savanna). For instance, encountering a jungle temple prompts the player to decide whether to enter or not. The outcome of the mini-game affects the player's overall progress in the main game.

Technical Specifications:

1. Global Variables:

- The program initializes global variables such as `firstTime`, `northCounter`, `southCounter`, `eastCounter`, `westCounter`, `travelCounter`, and `environment` to keep track of game state and player progress.

2. Screen Clearing:

- The `clearScreen()` function is defined to clear the console screen, providing a clean interface for the player. It uses the `os.system` command to clear the screen depending on the operating system so that it works on Linux, Mac and, Windows.

3. Player Input:

- The `getplayerInput()` function retrieves the player's input, handles the first-time output, and updates the `travelCounter`. It also dynamically changes the game environment from jungle to other environments after a certain number of travels.

### this has info about split up files still unsure what the verdict on splitting is for this proj check back

4. Direction Outputs:

- Directional outputs are stored in the `directionOutputs` module, which is imported from `jungle\_lists.py`. The outputs provide descriptive messages corresponding to each direction.

5. Command Processing:

- The `getCommand(playerInput)` function processes the player's input, converts it to lowercase, and determines the appropriate action based on the input. It triggers the mini-game with a 1 in 10 chance and updates the directional counters.

6. Mini-Game:

- The `miniGame(environment)` function initiates a mini-game based on the environment (currently jungle or savanna). It prompts the player to make a decision (enter or not) and presents a challenge with potential consequences. The outcome influences the main game progression.

7. Main Loop:

- The `main()` function initiates the main game loop. It prompts the player with initial instructions and continues until the player decides to quit or reaches a travel limit. The game over message is displayed with ASCII art.

8. Easter Egg:

- An easter egg is implemented where entering a specific command ("supersecretsecret") triggers a special message, adding a hidden element to the game.

9. Testing and Error Handling:

- The program includes error handling for non-string and non-integer inputs. It also provides visual cues for the player's choices and feedback messages in case of invalid input.