**Comp 120-01**

**Final steps 3 & 4**

**Curse of the Jungle**

Luke Colvin

Devin Kagak

Olivet Nazarene University

Professor Kender

December 11, 2023

Functional specifications:

Explore:

The name of our game is The Curse of the Jungle! This program is a choice-based adventure game where the player can choose to travel the directions north, east, south, and west. The theme we set is based in a jungle environment but while traveling you have a number that follows you wherever you head indicating how far you go into the never-ending jungle. As you head in there is a glimpse of different biomes, but you will find that you can never reach it because the distance gets longer with each step.

Randomized mini game encounters:

As you explore you will randomly find that there are mini games within our game! One being the unescapable jungle temple (exaggerated) where you are placed at the base of the jungle temple and given the choice of entering 3 doors, each door holds weight because there are up to 2 death doors with a minimum of 1 safe door leading to safety to continue your adventure. There is another mini game where you are given a tablet and a creature beckons you to play his game, this game is hangman. The hangman game consists of 5 jungle themed words to try to guess if you don’t you will not get the satisfaction of beating the taunting creature.

Goal:

While you are finding your way through the jungle you are also given the choice to quit and end the game. The player's goal is to explore the jungle, find minigames and travel as far as possible without perishing. The game keeps track of the number of times the player travels in each direction to trigger discovery of environments.

Technical Specifications:

1. Global Variables:

- The program initializes global variables such as `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.

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. 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.

Luke Colvin – I spent approximately 15 hours working on this project the issues i encountered had to deal with using global variables before finding out we were not supposed to use them, so I had to rebuild most of the game taking out the global functions some of the original functionality was sacrificed but we no longer have global variables. I replaced the global variables with a class that I was unable to transfer all the functionality to. I plan to work more in both local variables and classes to get better acquainted with them.

Devin Kagak – I have spent about 10 hours making mini games and learning many types of code. The errors I have encountered where functions are not calling because I have entered some wrong names.

















