|  |  |
| --- | --- |
| Full Name | Ethan D’Souza |
| Student ID | 1735988 |

Final design

1. **Introduction**

* Input: Player's name
* Output: Personalized greeting using the player’s name and a description of the initial setting.
* Story: The player is standing at the entrance of a mysterious jungle, with two paths in front of them.

1. **First Decision: Path Choice (String-based decision)**

* Input: Choice between 'forest' or 'cave' (string input)

Decision 1:

1. If the player chooses 'forest':

* Output: Describes entering a dense forest and hints at future challenges.
* Continue: Go to Decision 2 (number-based decision).

1. If the player chooses 'cave':

* Output: Describes entering a dark cave with eerie surroundings.
* Continue: Go to Decision 3 (float-based decision).

1. **Second Decision: Forest Path (Integer-based decision)**

* Input: Integer value (choice of 1, 2, or 3 representing different doors in the forest).

Decision 2:

1. If the player chooses Door 1:

* Output: Leads to a peaceful clearing.
* Continue: Go to Decision 4 (string-based decision).

1. If the player chooses Door 2:

* Output: Encounters an obstacle (wild animals).
* Continue: Use a Boolean operator to decide the outcome.

1. If the player chooses Door 3:

* Output: Leads to a treasure chest.
* Continue: Ask the player if they want to open it (yes/no decision).

1. **Third Decision: Cave Path (Float-based decision)**

* Input: Float value (number of meters to jump across a gap in the cave)

Decision 3:

1. If the player chooses to jump a distance less than 5 meters:

* Output: They fall short of the gap and need to find another way out.
* Continue: Move to another decision path.

1. Otherwise-if the player chooses between 5-10 meters:

* Output: They make it across the gap.
* Continue: Proceed further into the cave.

1. Otherwise-if the player chooses more than 10 meters:

* Output: They overshoot and land in danger.
* Continue: Use Boolean operators to determine if they can escape the danger.

1. Otherwise (if the input is not a valid float)

* Output: “Please enter a valid number, to jump across.”
* Loop back to ask for input again.

1. **Fourth Decision: Encounter (String-based decision)**

* Input: Choose a potion (red, blue, or green).

Decision 4:

1. If the player chooses 'red' potion:

* Output: Grants strength.
* Continue: Use Boolean logic to handle further challenges with enhanced abilities.

1. Otherwise-if the player chooses 'blue' potion:

* Output: Grants invisibility.
* Continue: They can sneak past enemies.

1. Otherwise-if the player chooses 'green' potion:

* Output: Heals player form previous damage.”

1. Otherwise, else (if input is not either of 3 colors)

* Output: “That is not a valid potion, please try again.”
* Loop back to ask for input again.

**5. Boolean Operators in Decisions**

Used in multiple decisions:

1. In Decision 2 (Door 2: encounter with wild animals):

Boolean operator checks if the player has the right equipment to handle the animals.

1. In Decision 3 (Cave jump):

Boolean operators evaluate if the player’s jump is successful (e.g., if distance >= 5 and distance <= 10).

1. In Decision 4 (Potion):

- Boolean operator checks if the potion grants abilities to pass future obstacles.