**AI BASED INTERACTIVE STORY TELLING**

**INTRODUCTION:**

"Adventure of the Lost Forest" is an interactive text-based adventure game built using Python and Pygame, where the player navigates a mystical forest by making decisions that influence the progression of the story. The player is presented with various choices at key points in the game and these decisions determine the outcome of the game. The game ends when the player reaches one of the possible conclusions, with the option to restart or quit.

**AI-ALGORITHM INVLOVED:**

1. **Choice Tracking:**

The game primarily utilizes decision-making AI algorithms. The "AI" here is a simple choice-tracking mechanism that influences the game state based on the player’s previous decisions. Here's how the AI works like:

1. **State Transitions:**

Based on the player's choices, the game transitions between different states (e.g., from the forest to the lake or the exit). The player's actions are mapped to specific game outcomes, and the game state changes accordingly. This is a simple form of decision-tree logic.

1. **Decision-making Logic:**

The game simulates decision-making by presenting options and responding based on the player’s input. While not truly "intelligent," the program uses predefined rules to determine the flow of the narrative. It is an example of rulebased decision systems.

**TOOLS AND TECHNOLOGIES:**

* Python
* Pygame Library

**FLOWCHART:**

Start

|

v

[INTRO: Enter Forest? (Y/N)]

| |

| v

| [GAME OVER]

v

[FOREST: Follow Light? (Y/N)]

| |

v v

[LAKE] [EXIT PATH]

| |

| v

| [GAME OVER]

v

[LAKE: Drink or Talk? (1/2)]

| |

v v

[MAGIC] [KNOWLEDGE]

| |

v v

[GAME OVER]

A diagram of a forest

AI-generated content may be incorrect.

**CODE SNIPPETS:**

**1.State Management**

INTRO, FOREST, LAKE, MAGIC, KNOWLEDGE, EXIT\_PATH,

GAME\_OVER = 1, 2, 3, 4, 5,6,7

current\_state = INTRO

choices = []

**2.Drawing Text on Screen**

def draw\_text(text, color, y\_offset=0):

text\_surface = font.render(text, True, color)

text\_rect = text\_surface.get\_rect(center=(WIDTH // 2, HEIGHT // 2 + y\_offset))

screen.blit(text\_surface, text\_rect)

**3.Intro Scene Logic**

def handle\_intro():

screen.fill(CYAN)

draw\_text("Welcome to the Lost Forest!", BLACK, -50)

draw\_text("Press 'Y' to enter or 'N' to leave.", BLACK, 50)

**4.Main Story Branching**

if event.type == pygame.KEYDOWN:

if current\_state == INTRO:

if event.key == pygame.K\_y:

choices.append("adventurous")

current\_state = FOREST

elif event.key == pygame.K\_n:

current\_state = GAME\_OVER

**5.Restarting the Game**

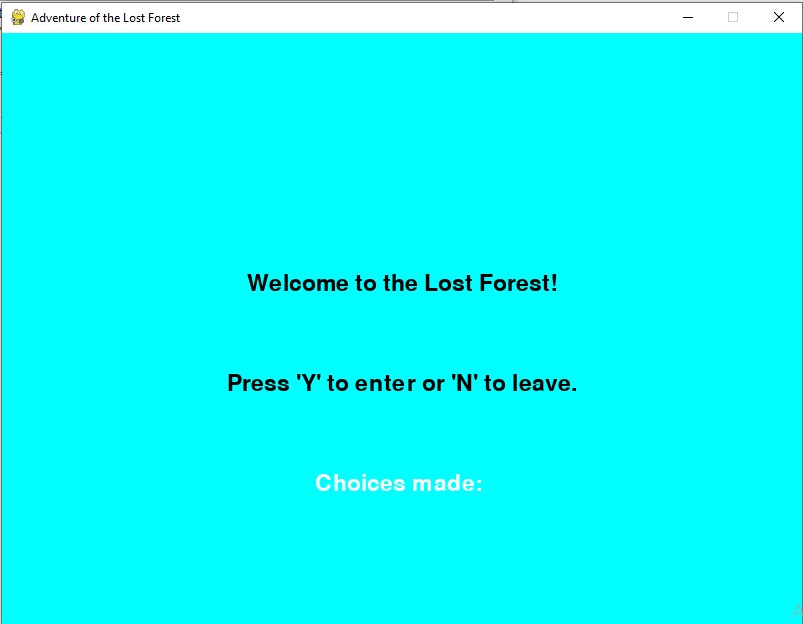
def restart\_game():

global current\_state, choices

current\_state = INTRO

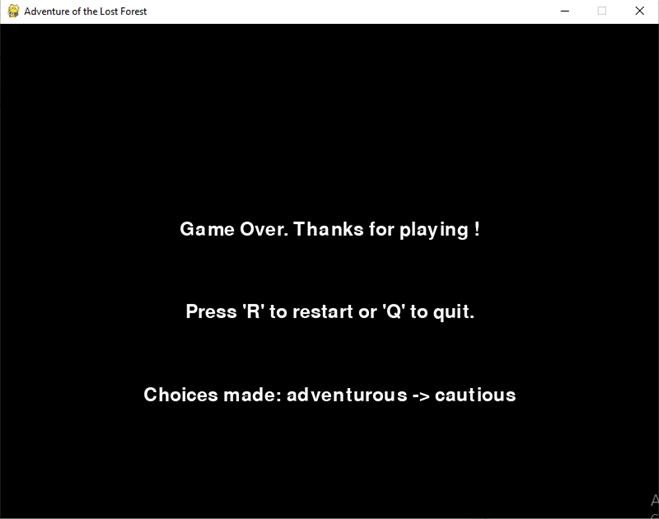
choices = []

**OUTPUT:**



A green screen with white text

AI-generated content may be incorrect.



**CONCLUSION:**

The “Adventure of the Lost Forest” mini-project successfully demonstrates how interactive storytelling can be brought to life with Python and Pygame. By structuring the narrative around clear game states and branching decisions, the project:

1. **Reinforces fundamental game-development concepts**

Event loops, keyboard input handling, and screen updates.

**2. Illustrates state-driven storytelling**

Player choices (“adventurous” vs. “cautious”) visibly alter the plot, providing a hands-on example of branching logic.

**3. Blends creativity with code**

Simple visuals and text create an engaging experience without heavy assets.

Through building and testing the game, I deepened my understanding of Pygame’s core workflow and the design of decision trees for narrative flow.

Future enhancements could include adding images and sound effects for richer immersion, introducing additional story branches and puzzles to extend replay value, and implementing save/load functionality so players can continue their adventure later. Overall, this project proved to be an effective and enjoyable way to combine programming skills with creative storytelling.