Tap and Tale: Story Book Program

By: Sonia Siddiqui

We Will Describe:

01

Our Code

Data Structures and Algorithms Used

02

Results

03

Model and Implementation

04

Running our Software

How users can interact and create their own story

CLASSES USED

InteractiveStoryTree

Represents a single story, containing a series of connected narrative nodes in the tree-like structure.

StoryNode

Represents a decision point or event in the story, containing options to the possible choices and how they lead to each other.

Methods in InteractiveStoryTree Class

def__init__(self, start_node): Constructor method that initializes an instance of InteractiveStoryTree. It takes a start_node parameter, representing the starting node of the story.

def display_current_node(self): Method to display the text of the current story node along with the available choices.

Methods in InteractiveStoryTree Class

def make_choice(self, choice_index):
Method for the player to make a choice by providing the index of the choice.It uses breadth-first search (BFS) to traverse the story tree efficiently.

def end_story(self):
Method to handle the end of the story.It prints a congratulatory message to the player for completing the story.

Methods in StoryNode Class

def __init__(self, text, choices=None)
Constructor method that initializes a StoryNode instance with the provided text and optional choices.

def add_choice(self, choice, next_node):
Method to add a choice to the current story node.
It allows the story to branch out based on the choice made by the player.

More data structures

Dictionaries

Used to store the story data efficiently, with nodes indexed by unique identifiers.

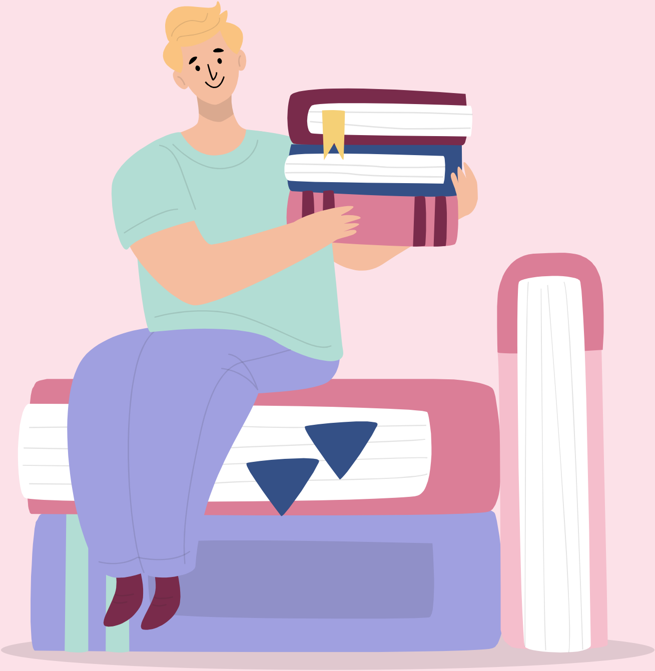
Stacks (Queues)

The queue maintains a collection of nodes to be explored during the breadth-first search (BFS) traversal of the story tree.

Algorithm

BFS Algorithm

To traverse the story graph and navigate between nodes based on user choices,
a breadth-first search (BFS) algorithm to explore the story paths.
This BFS approach guarantees that the traversal progresses from the starting node (initial story node) to the end nodes (terminal nodes) in a breadth-first manner, providing an organized and comprehensive exploration of the story's narrative paths.



01

The Results

Console Test Run

```
Hello, fellow explorer! Welcome to Tap and Tale where you can
go on a journey to choose your own story!
Choose a setting aligned with a character!
1. You are a motorcycle racer in New York City
2. You are a spacecraft pilot in Tokyo, Japan
Enter your choice: 1
This or that?
1. Become a video game character in the city.
2. Become a villain in the city.
Enter your choice: 2
You are now a villain. Choose your first evil act.
1. Steal a valuable artifact from the museum.
2. Cause chaos in the city streets.
Enter your choice: 2
The authorities catch you and you are given life imprisonment.
```

01

Model and Implementation

IMPLEMENTATION

- Story nodes are defined using dictionaries, with each node instantiated with a unique text and choices.
- Choices are connected to subsequent nodes using the add_choice() method.
- The interactive story tree is initialized with the starting node.
- The user begins the story by interacting with the initial node, making choices to progress.
- The program continuously prompts the user for input until reaching a terminal node, signifying the end of the story.

USER INTERACTION

- Intuitive for the user
- Story options bounce off of previous options based on nodes

01

Running the Code

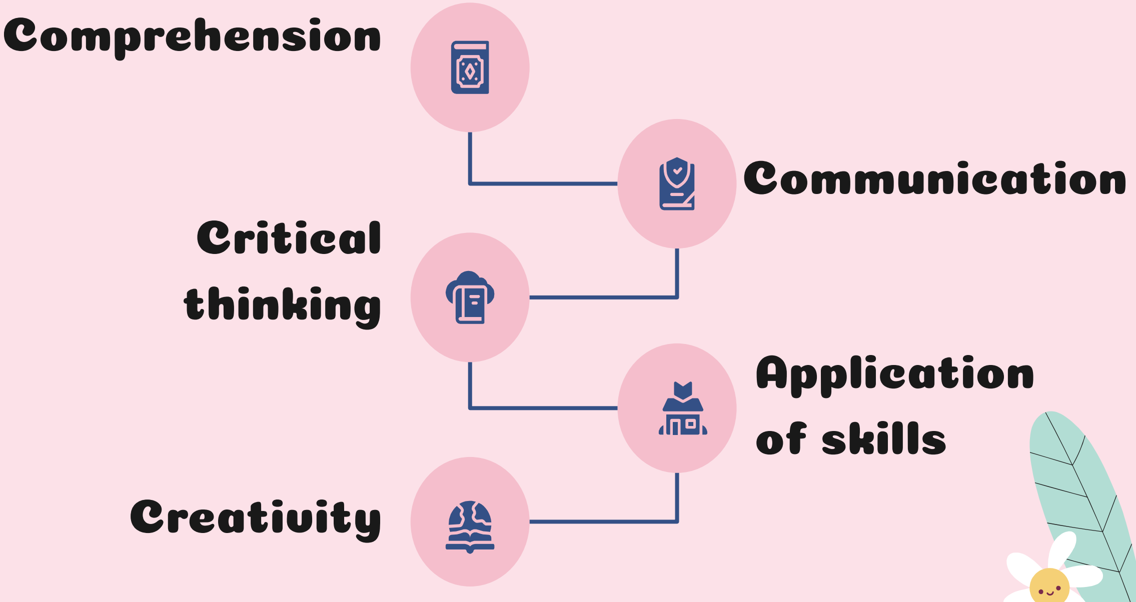
How users can interact and create their own story

Test Cases

1. Invalid Input Handling Test ✔️
2. Traversal of Story Tree Test ✔️
3. Terminal Node Handling Test ✔️

Discussing the Results

- The program prompts the user to choose between two options.
- Each option leads the program to prompt another option, which all uniquely create a story
- **This program can help encourage children who struggle with creativity and writing skills to write their own story in a fun and easy way!**


Comprehension

Communication

Critical thinking

Application of skills

Creativity

References

1. Real Python Team. "Python Hash Table." Real Python, Real Python, https://realpython.com/python-hash-table/.
2. Price, Dan. "Python Text Adventure Game: Create Your Own with These Tools." MakeUseOf, MakeUseOf, https://www.makeuseof.com/python-text-adventure-game-create/.
3. Pierian Training. "BFS (Breadth First Search) Implementation in Python." Pierian Training, Pierian Training, https://pieriantraining.com/bfs-breadth-first-search-implementation-in-python/#:~:text=BFS%20starts%20from%20a%20source,vertices%20and%20their%20adjacent%20vertices

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**