Narrative Quest Generation

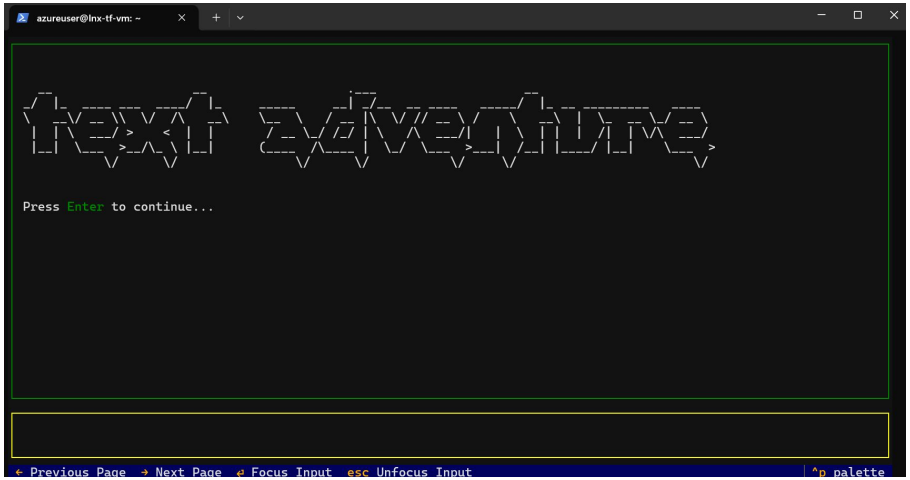
By: Joshua Lester & Caleb Stewart

Project Reminder

Inspired by games like Zork and AI Dungeon, our goal is to create a text-based adventure game where the story evolves dynamically based on the player's actions. We're combining language models and vector search to make the game more flexible and immersive.

Interface

- Built using Python's Textual (a popular TUI framework)
- Runs the application in the terminal
  - Chose this because we SSH into an Azure VM equipped with a good GPU
- Still a work in progress
  - Would like to allow application choose the story to be taken
  - Couple of bugs here and there



Language model

We're using the Pygmalion-3-12B language model with Ollama for story generation.

- Prompt engineering:
  - We're actively working on prompt engineering to ensure the model stays immersed in the narrative and responds consistently with the evolving story.
- Retrieving context
  - The retrieved memory is also appended to the prompt
  - Context is stored in a vector database

Context Retrieval

Using for FAISS to store context and retrieve it when necessary. The goal of this is to let the model “remember” important past events so it can generate coherent story responses over time.

This is done by using FAISS (Facebook AI Similarity Search)

Context Storage Process:

1. Player inputs a command
2. Embed the input using a SentenceTransformer
3. Query FAISS for the most similar past events
4. Add those events to the prompt
5. Generate response with language model
6. Summarize + embed the response, then add it into FAISS

Need to do

Identify player intent and relevant objects using natural language processing

Maintain dynamic state tracking for environment and inventory