Adventure Quest: An Interactive Game with Factor Graph Decision Making

Maisha Rahman, Mohona Haque, Ruhama Fabiha Rahman, Sheik Mehedi Hassan
Department of Electrical and Computer Engineering,
North South University, Dhaka, Bangladesh

Supervisor:Mohammad Shifat-E-Rabbi
Department of Electrical and Computer Engineering,
North South University, Dhaka, Bangladesh
Email: rabbi.mohammad@northsouth.edu

***Abstract*—This paper presents the design and development of *Adventure Quest: AI Chronicles*, an interactive storytelling game enhanced by a decision-support AI system based on factor graphs. The game incorporates narrative immersion with player agency and integrates artificial intelligence to recommend optimal decisions. The AI uses belief propagation to infer actions based on the player's energy and reputation attributes. This work demonstrates the feasibility and engagement potential of integrating probabilistic graphical models in narrative-based game design.**

***Index Terms*—Interactive Fiction, Factor Graph, Game AI, Tkinter, Python, Belief Propagation, Decision Support, Storytelling**

I. Introduction

Artificial intelligence is increasingly being adopted in digital games to drive decision-making, player adaptation, and dynamic storytelling. In traditional narrative-driven games, decisions are often static and predetermined. This project explores how a probabilistic AI model can dynamically suggest player actions to enhance the decision-making experience in interactive fiction.

The objective of this work is to create a cinematic and immersive storytelling game where each decision influences the narrative outcome. The core contribution is the integration of a factor graph-based AI engine that infers the most probable optimal action at each story node based on real-time player attributes.

II. Methodology

A. System Architecture

The game was developed using the Python programming language with the following primary components:

- **Tkinter GUI:** Used to render the game interface, including background images, text, buttons, and log display.
- **StoryManager:** Loads story nodes from a JSON file where each node contains text, background image, audio, and player choices.
- **GameState:** Maintains the current state of the game including the player's name, energy, reputation, inventory, and current story node.
- **AIDecisionEngine:** A factor graph model implemented with `pgmpy` to infer the best action for the player.

B. Game Flow

The gameplay follows a structured sequence:

1) Upon launch, the player is prompted to enter their name.
2) The story begins at the root node labeled `start` and progresses through linked nodes based on the player's choices.
3) At each story node:
   - A background image and ambient music are presented.
   - The narrative text and available choices are displayed.
   - The AI engine evaluates the player's current **energy** and **reputation** values and recommends the most suitable action.
4) Upon making a choice, effects are applied (e.g., decreasing energy, increasing reputation, adding items).
5) The system transitions to the next node, repeating the process until an ending node (with no choices) is reached.

C. AI Decision Engine Using Factor Graphs

The AI component uses a probabilistic model to suggest decisions. The factor graph is composed of three random variables:

- E: Energy Level (Low, Medium, High)
- R: Reputation Level (Low, Medium, High)
- A: Action (Rest, Sneak, Fight)

The conditional probability distribution $P(A|E,R)$ is computed using belief propagation. The AI selects the action with the highest posterior probability.

$$P(A = a \mid E = e, R = r) = \frac{P(E = e, R = r, A = a)}{P(E = e, R = r)} \quad (1)$$

This inferred action is then matched to the closest available textual choice at the story node.

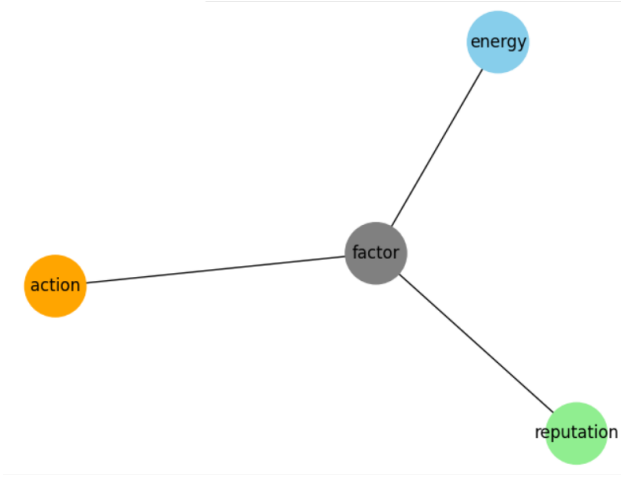

Fig. 1: Factor Graph

D. User Interface and Experience

The UI incorporates dynamic visual and audio elements:

- Background images with a fade-in transition enhance visual storytelling.
- Sound effects and background music are mapped per scene for immersion.
- The interface displays player statistics and a scrollable story log for narrative tracking.
- AI suggestions are shown to assist but not override player decisions.

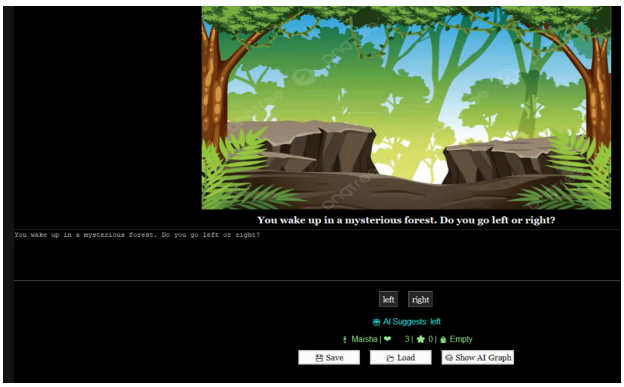

Fig. 2: Game interface with AI suggestions and narrative content (placeholder)

III. Game State Model

The internal game state is maintained through the following parameters:

- **energy:** Integer from 0–3, mapped to low/medium/high.
- **reputation:** Integer from 0–3, mapped to low/medium/high.
- **inventory:** List of collected items (e.g., `["sword", "map"]`).
- **name:** String input from the player.
- **current_node:** String identifier of the current story node.

The AI's theoretical state space is:

$$\text{Energy (3 levels)} \times \text{Reputation (3 levels)} \times \text{Actions (3 types)} = 27 \text{ combinations}$$

IV. Challenges

Throughout development, the following challenges were addressed:

- **AI Mapping:** Translating abstract AI decisions to descriptive in-game choices required approximate semantic matching.
- **Fade-In Effects:** Implementing image transitions required maintaining alpha-level interpolation with PIL and Tkinter.
- **Data Synchronization:** Ensuring consistent state updates across AI, UI, and gameplay logic was crucial for reliability.
- **Save/Load Mechanism:** Game state was serialized using `pickle`, but robust exception handling was required to prevent crashes from corrupted or missing files.

V. Conclusion

Adventure Quest: AI Chronicles illustrates how probabilistic reasoning can enhance interactive storytelling. The integration of a factor graph-based AI provides context-aware decision support, improving player engagement without compromising narrative freedom. The project's modular architecture allows for easy expansion in terms of story complexity, state variables, or AI models, paving the way for more adaptive and intelligent game experiences.