

Objective

Build a multi-agent, multi-turn game application using LangGraph that demonstrates modular agent logic, step-by-step orchestration, and session-level tracking.

Note: The focus is not on building a polished UI. The primary objective is to evaluate how effectively the AI agents are structured, state is managed, and game logic is executed through LangGraph.

System Architecture

1. Frontend (Web UI)

- Presents a basic interface to:
 - Show the **main menu**.
 - Accept and display user interactions during gameplay.
- Can be CLI-based or a minimal web UI.
- UI must support:
 - Game selection input
 - Question-answer interactions
 - Game result messages
- **UI is not the focus.** It should be functional, not fancy.

2. LangGraph Application (Core Logic)

- Implements a **multi-agent architecture** with:
 - **Game Selector Agent** to route the flow.
 - **Number Game Agent** using binary search logic.
 - **Word Game Agent** using clue-based narrowing and guessing.
- Orchestrates state transitions and maintains agent-specific state.
- Tracks how many games were played in the current session.

3. Database Layer (Optional)

- Optional integration to persist:
 - Game history
 - Total number of games played
 - Win/loss counts
- Can be simple (SQLite) or cloud-based (e.g., Supabase).

Game Modes

Number Game

- Prompts the user to think of a number between 1–50.
- Uses binary search logic (e.g., “Is it greater than 25?”).
- Continues narrowing until the correct number is guessed.
- After success:
 - Congratulate the user.
 - Increment the number game counter.
 - Return to the main menu.

Word Clue Guesser

- Prompts the user to choose a word from a fixed list:
`["apple", "chair", "elephant", "guitar", "rocket", "pencil", "pizza", "tiger"]`
- The agent asks up to 5 descriptive yes/no/maybe questions.
- Then attempts to guess the word.
 - If correct → Show success message → Return to main menu.
 - If incorrect → Ask if the user wants to retry or return to the main menu.
- The word game counter must increment after each attempt.

Main Menu Flow

- Shown at app start and after every game:

markdown

Copy

Edit

Choose a game:

1. Number Game

2. Word Game

(Leave blank to exit)

- If input is **blank**, the app responds:

arduino

Copy

Edit

You played X word games and Y number games.

Key Evaluation Criteria

- Correct implementation of LangGraph agents and routing.

- Multi-turn logic handled per agent with session awareness.
- Accurate stat tracking (per session).
- Functional return to menu and clean exit.
- Optional DB handling is a plus but not mandatory.
- The **quality of AI agent design and LangGraph orchestration** is the core of this task.