Workshop Instructions – Part 1: Build a GuidedConversation

Overview

In the first part of this workshop, you'll be building out the `GuidedConversation` model to simulate a realistic conversation between personalities of your choice.

Currently, the project includes two participants (`participantOne` and `participantTwo`) but **nothing that defines the actual conversation**. Your task is to design the schema, add properties, and use `@Generable` and `@Guide` annotations to shape the dialogue.

Your Tasks

1. Extend `GuidedConversation`

- Define how the conversation should unfold.
- Decide:
  - How many turns or stages should there be?
  - Should the conversation be a fixed length or flexible?
  - Do you want to model each exchange with a new `@Generable` type (e.g., `ConversationExchange`)?

> 💡 Remember: **Generable models are nestable**. You can create smaller Generable types for exchanges or turns and reference them in your main schema.

---

2. Use `@Guide` Effectively

- Add descriptions that explain *how* each property should be generated.
- Examples of constraints:
  - `.count(n)` → number of items in a list
  - `.range(1...10)` → numeric ranges
  - `.anyOf([...])` → select from a fixed set
- Be specific: use tone, format, and style guides to control the output.

---

3. Map Your Schema to the UI

- Conform your updated `GuidedConversation.PartiallyGenerated` to `GuidedConversationDisplayable`.
- Implement `conversationExchanges` by mapping your properties into `DisplayableConversationExchange` objects.
- The SwiftUI UI is already wired up — once you return values here, they'll display in the app!

---

4. Update `WorkshopConstants`

The `WorkshopConstants.swift` file controls how your conversation is generated. You can adjust three key areas:

- **Instructions**
  High-level rules for the model to follow (e.g., *"Simulate conversations between the provided participants. Use the image lookup tool to get image URLs"*).
- **Prompt**
  Defines the scenario and participants. Example:
  ```swift
  static let prompt: String = "A conversation between Obi Wan Kenobi and George Washington"
  ```
- **Sampling Mode**
  Controls creativity vs determinism:
  - `.greedy` → deterministic, always the same output.
  - `.temperature(Double)` → adds variability, higher values = more creative.

> ✨ Experiment! Try swapping personalities, rewriting the prompt, or changing sampling modes to see how the output changes.

Notes

- Order matters: properties are generated top-to-bottom, each depending on previous context.
- Don't worry if your model feels simple at first — you can refine and expand it as you go.
- ✅ Everything you see in the slides (comments, TODOs, ASCII owls 🦉) is also in the project.

Optional Stretch Goals

- Extend `Personality` with richer details (tone, humor, formality).
- Support **n participants** instead of just two.
- Add constraints that force interesting styles (e.g., “in Yoda’s voice” or “debate style”).

Next Step

Once your `GuidedConversation` schema is complete, mapped to the UI, and paired with updated constants in `WorkshopConstants`, run the project and view your generated conversation transcript in real time!