Who Wants To Be A Millionaire

A voice-operated game by Maria Szawerna

LT2216 Dialogue Systems, University of Gothenburg

The Game

- Based on the popular TV game show franchise
- The player is a participant in a re-enactment of the game
- Difficulty levels
- 12 (practically 13) questions with 4 answers each
- 2 lifelines
- Safe points and ability to leave after a successful answer
- Chit-chat with the game host

Click to play Who Wants to be a Millionaire!

The Technicalities

- Based on the dialogue-based apps from the labs
- XState, JavaScript/TypeScript, Microsoft Azure
- Open Trivia Database API

Trivia API

The Open Trivia Database provides a completely free JSON API for use in programming projects. Use of this API does not require a API Key, just generate the URL below use it in your own application to retrieve trivia questions.

All data provided by the API is available under the Creative Commons Attribution-ShareAlike 4.0 International License. 

The Challenges

- Understanding XState peculiarities and programming in JS/TS:
  - A lot of documentation reading, JS tutorials, trial and error
- Transforming the data from the API into a format without fixed answer patterns:
  - Using an implementation of a shuffling/scrambling algorithm together with lists of all answers and a list of correct answers
- Implementing varied dialogue:
  - Using lists of utterances with varied expressions, pre-generated with some information (the player's name, questions and answers)
- Abstracting away instead of copy-pasting:
  - Once more, using arrays, current question counters, and indexing to change the content in the general frame of how a question and chit-chat plays out
- Implementing lifelines:
  - Choosing the only two of the actual lifelines used in the show that do not involve other humans

The Course Contents

- XState-related labs:
  - Being able to “translate” the game show formula into a state machine and implement it
  - Understanding how to use APIs
- “Dialogue theory” lectures:
  - Understanding the importance of grounding and other strategies, as well as the variation in the dialogue
- TTS lab:
  - Being more familiar with Azure’s voice options and being able to select a host voice

The Future

- More randomness in host utterances
- Implementing elements such as the current amount of money or question number together with the aforementioned issue
- Deviating from the game show's format and allowing the users to select the number of questions they want to answer or letting them select the categories or change the timer on answers
- Better user interface, potentially with the rules written out, a timer ticking down while the answers are awaited, color cues when answers are given (red and green)

Thank you for your attention!

And now it's time for the demo...