

# Project presentation

# Dialogue systems LT2216

Spring 2022

**Max Boholm**

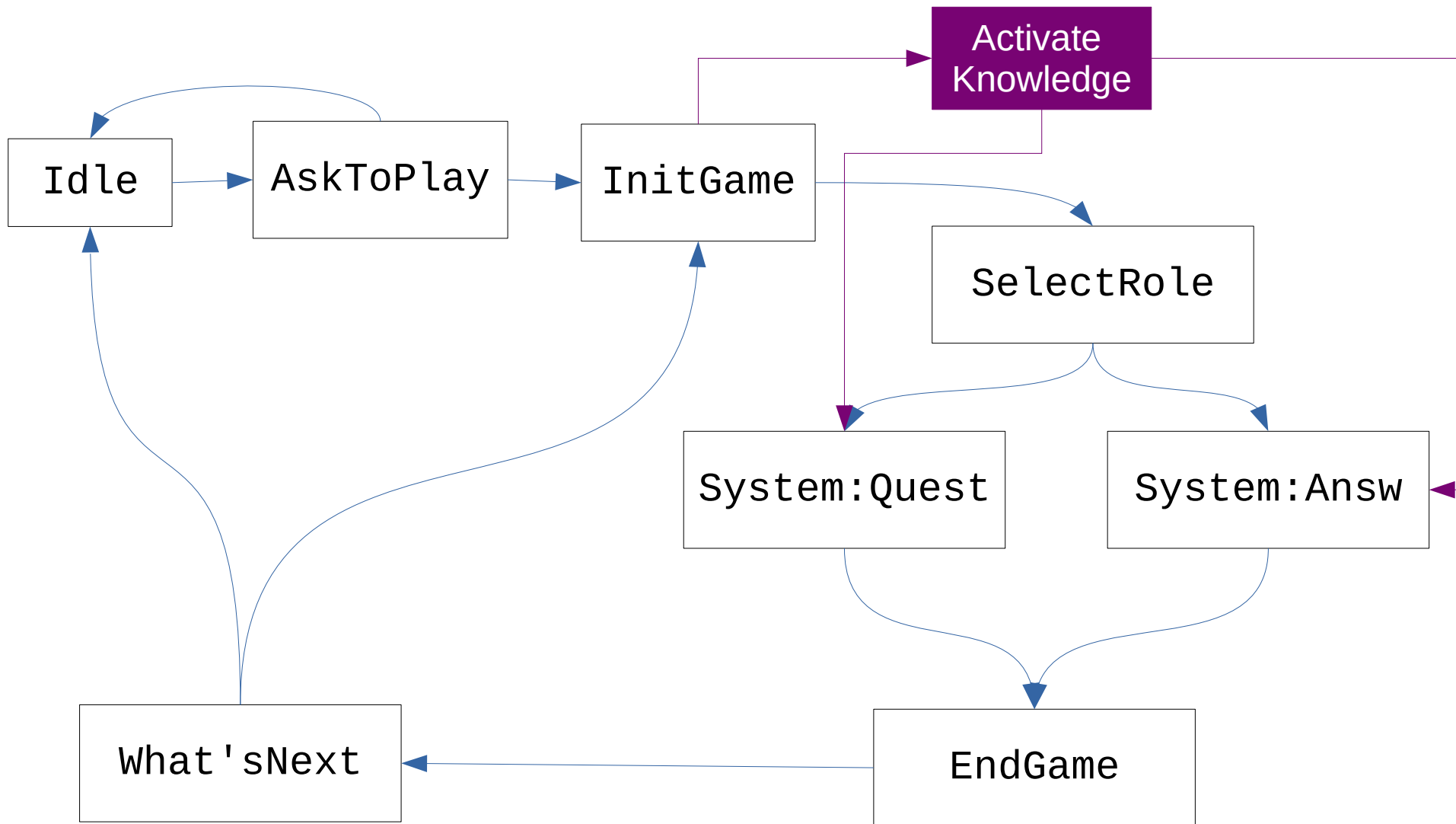
# Game: 20 questions

- A two-player game where one is assigned Answerer (**A**) and the other Questioner (**Q**).
- **Q** decides about a character **C** (without revealing whom).
- **A** is allowed to ask 20 Y/N questions about **C**, which than are answered (truthfully) by **Q**.
- The game terminates by **A** successfully guessing the identity of **C**; or **A** runs out of attempts.

# Implementation

- Implemented in *XState*: Typescript/Javascript library for statecharts
  - Based on the statechart formalism in Harel (1987) and State Chart XML (SCXML)
- TTS and ASR implemented through Microsoft Azure Speech Studio
- Speech Synthesis Markup Language (SSML) for a few system utterances
- Overall structure from course labs (e.g., React environment, index file)

# Overall structure



# System as Questioner

- From known features  $F$  randomly select (pop)  $f$
- Ask if character has  $f$  and record user response ( $v$ )
- Add object  $\{f : v\}$  to `buildup` (array)
- Decision:
  - Compare `buildup` and `knowledge`
  - If unique character given `buildup`, then `guess` = unique character
  - If no unique character and attempts left, ask of new feature  $f'$
  - If no attempts left, pick random guess from possible characters (given `buildup`)

# System as Answerer

- Randomly select character  $\mathbf{c}$  from known characters  $\mathbf{C}$
- Wait for user question  $\mathbf{q}$  (about  $\mathbf{c}$ )
- Extract feature  $\mathbf{f}$  from  $\mathbf{q}$  (rule-based `qParser()`)
- System answer = `knowledge.c.f` (json structure); give answer
- While attempts left: invite new question  $\mathbf{q}'$

# Challenges and features

- Game-related challenges
  - implement structure of game (System as A / Q)
- Recognizing and parsing user utterances
  - Grammar; `qParser()`
- Respond to “out-of-grammar” utterances
  - Identify and respond to “out-of-grammar” user utterances; after *n* attempts abort game
- Timeouts
  - Re-state system utterance after no input from user; after *n* timeouts, abort
- AttemptsLeftChecking feature
  - Keyword “Attempts” (from user), gives info about attempts left; history node in statechart
- SR Confidence Level
  - If confidence level of ASR is below threshold *t*, ask for clarification “Did you say ...?”

# Future work

- Extending the knowledge (obvious, but perhaps not that interesting)
- Parsing user utterances (machine learning for NLU, perhaps use Rasa)
- Making a smarter questioner: There is a logic for 20 questions! (features are related)
- Making a more accurate answerer: there is semantics in knowledge! (Again, features are related: [male, a man, female, a woman]. Use WordNet?)
- User initiative (S: do you want to be questioner or answerer? U: I have decided on a character)



Thank you!