**Introduction**


Allows children to create their own conversation agents through which they can share stories and experiences

Helps children get a better sense of self

Helps children start thinking like programmers as they have to formalize their thoughts into a conversation tree resembling the one below

**Requirements**

Allow multiple interfaces, both standalone and Zora, a multi-user graphical environment specifically developed to help users design and inhabit a virtual city (see other DevTech posters)

Everything must be visual, requiring no code entry by the user

**Sample Conversation Tree**

The system records the response as the name of the user.

The agent introduces itself and asks a question like “What is your name?”

Agent says “Nice to meet you name. How are you feeling today?”

User: I am sad.

User: I am happy.

Agent: “I am very sorry to hear that. Why are you sad?”

User: Lost dog.

User: I did bad on a test.

At this point the agent tells a story followed by “Do you feel better now?”

User: Yes. Thank you.

Agent: “I am sorry to hear that. Do you want me to tell you another story?”

User: Yes.

User: No.

**Key**

A Visited node.

Node not visited.

Path that was not taken. User input did not match the edge so the node to which it leads was not visited during the conversation.

Path taken between visited nodes. Note that the edges between nodes are user input.

**Conversation Trees**

Each conversation is represented as a tree of user inputs and responses

Nodes contain agent responses to user input

Edges are user inputs, that is an edge is followed is the user input matches the pattern provided for that edge

When the user enters text, the system searches through the edges from the current node. If a match is found it follows that edge to the next node, otherwise it follows the specified default node

User-specified variables are collected at the edges

For example, if the user says “My name is Andrew.” and the creator of the conversation specified to extract the name at that node the system will store “Andrew” as the name of the user

Creators can also label nodes as “story” nodes. When these nodes are reached the system analyzes the conversation up to that point and picks the best story added by the user as the response

**Current Work**

Working on an attractive and easy-to-use Flash interface

Enhancing the conversation algorithm to allow for better response handling

**Future Work**

Expanding the number of matching options, allowing “smarter” agents to be created

Expand the algorithm to account for misspelled words and synonyms

For example, “cheerful” would be matched by both “cherful” and “cheerfull”

Different weights for keywords and matching conditions based on modifiers

For example, “happy” would receive a score of 3 and “very happy” a score of 7