

# Programming by Voice in Alice

George Tankersley  
(Mentor: Dr. Jeff Gray)

Demonstrating the potential to teach computer science to young children with physical disabilities.

## Project

### Abstract

The main goal of this research project is to empower software developers who are physically disabled (e.g., those with limited limb mobility) with tools that will allow them to participate as equal partners and contributors in the information society. This is not a trivial problem in a world where desktop computing is dominated by graphical user interfaces (GUIs) driven by point-and-click metaphors. Software development tools are influenced strongly by such interfaces. The configurability of such tools to enable better access through speech interaction offers the potential to impact any category of user impeded by a point-and-click metaphor.

Alice is a programming language that provides a simplified environment that can serve as a first exposure to programming for students. Alice has been used in elementary and middle schools to introduce object-oriented programming in a fun way by creating video games and movies. In Alice, a programmer designs a world that contains various graphical objects that are programmatically controlled. However, due to its mouse-driven interface, Alice is fairly inaccessible to the physically disabled. The project goal is to add voice controlled functionality to the Alice environment. This was accomplished by integrating the CloudGarden Java Speech library into Alice. To further increase accessibility, scripting capability was added to allow even non-programmers to map chains of actions to particular words within the Java Speech grammar.

### Objective

The objective of this project is to create a voice-controlled interface that will allow a programmer to function in Alice without the use of a standard mouse and keyboard. The project provides an initial assessment of programming by voice to assist those with physical disabilities.

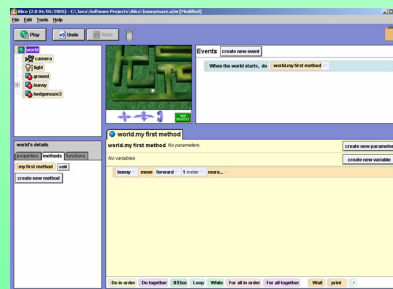
### Challenges

- Traditional voice command systems are unsuitable. They are mostly dictation systems and focus on typing but not mouse control.
- Alice is currently being rewritten – a good interface should still be adaptable to the next version without fundamental changes.
- The original Alice interface is highly intuitive. A non-conventional voice interface needs to match that ease of use.

## Background

### Alice

Alice is a project from Stage3 Research at Carnegie Mellon University. Its purpose is to provide a fun and non-frustrating first experience with programming concepts for elementary and middle school students. Programming with Alice is based entirely on a simple “drag and drop” interface. Students are taught “objects first” through creation of movies and video games.



Alice in use

### JSAPI

The Java Speech API is a standard created by Sun to define the interaction of Java with current speech recognition and synthesis technologies.

Recognition is driven by *grammars*, which tell the recognition engine what words to expect and in which patterns to expect them.

```
#JSPF V1.0;  
grammar AliceControl;  
public <openAlice> = [open] Alice;  
public <selectRecentTab> = [open] (Recent world);  
public <mostRecentWorld> = [open] ((World One) | (Most Recent [World]));  
public <playWorlds> = play [World];
```

Part of a Java Speech grammar file

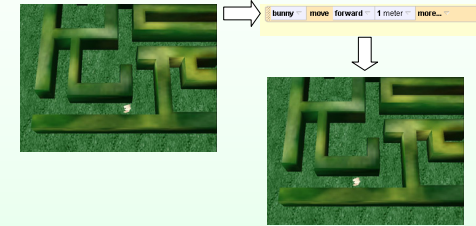


## Bunny in a Maze: Alice by Voice

### Step 1

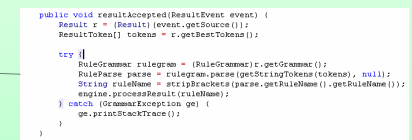
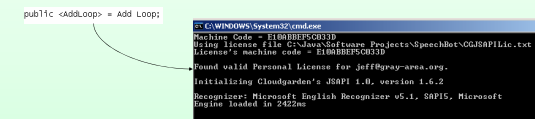
Consider a basic Alice world consisting of a rabbit trapped in a hedge maze. The rabbit has almost made it out of the maze, but its script does not quite reach the end.

The user has an objective: get the rabbit out of the maze.



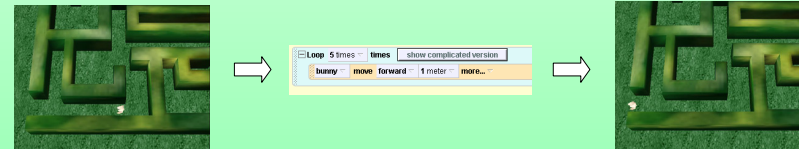
### Step 2

The user speaks the commands defined by the grammar to add a five-cycle loop to the script. The recognition engine passes the spoken words to the SpeechEngine class, which mediates between the standardized JSAPI classes and the custom Alice scripting.



### Step 3

With the new commands, the rabbit will move forward five times as far as it did before. Success!



## Future Work

- *More complete binding of the Alice interface:* Currently the project supports only the menus necessary to open a world.
- *Support for manipulating world objects:* It is not currently possible to add a new object to a world or to manipulate the positioning of an object already in a world.
- *Interaction with a running world:* The voice engine only allows you to watch a world and control its playback, but Alice worlds have the ability to be interactive.

## Acknowledgements

This project makes use of BeanShell, a lightweight Java scripting interpreter. BeanShell is dual-licensed under the GNU Lesser General Public License and the Sun Public License (JSR-274). Further information may be found at beanshell.org.