

Instituto Tecnológico de Costa
Rica
Curso: Programacion Orientada a
Objetos
Profesor: Mauricio Aviles



Propuesta de Proyectos #2 y #3

“2001Speak: Una interfaz de audio para el shell de Linux”

I Semestre
1 de mayo, 2015

Estudiantes:

Sara Castro Sáenz, 2014085332
Kenneth Callow Monge, 2013114955

Descripción general del proyecto

La ciencia ficción nos pinta un mundo donde los humanos hablan con máquinas con características profundamente humanas, a tal punto que nos superan en el intelecto y existen conflictos de poder entre nosotros. Este proyecto, titulado 2001Speak, busca llevar estas fantasías un paso más cerca a la realidad, dados los importantes avances en la tecnología de reconocimiento de voz, y la falta de herramientas de propósito general para comunicarse con las computadoras.

2001Speak consiste en una terminal de Linux implementada en Java que utiliza la librería en C PocketSphinx de Carnegie Mellon University, la cual interpreta las entradas audibles desde el micrófono del usuario a texto. 2001Speak interpreta estas entradas de texto como comandos a insertar y ejecutar en la terminal. Implementa la posibilidad de definir comandos personalizados por el usuario, los cuales consisten en palabras clave que se expanden en comandos más largos, secuencias de comandos, o en combinaciones de teclas. Por ejemplo, las palabras “bright 250” podrían convertirse en “echo 250 | tee /sys/class/backlight/intel_backlight/brightness”, lo cual pone el brillo en 25%; “close” podría convertirse en la combinación Alt-F4. Los comandos personalizados se crean en tiempo de ejecución, o sea, por medio de comandos de audio hablados por el usuario para ese propósito, por lo que le abre las puertas a todos los usuarios, aún poco técnicos, de “programar” sus máquinas con sólo hablar.

También se implementan los macros, los cuales consisten en “grabaciones” de todas las acciones que realiza el usuario, que son guardadas en un script de shell para cada una dentro de la carpeta de configuración de 2001Speak. El usuario indica cuándo comenzar a grabar y cuándo terminar, y un comando al cual asignarle el script. Esta herramienta, combinada con las anteriores, crea todo un sistema sobre el cual el usuario puede controlar su máquina, de una forma mucho más poderosa y mucho más amistosa que el uso normal de las interfaces de texto en Linux e interfaces en general. También es útil

para introducir conceptos de programación, ya que los hace conceptualmente más accesibles y tangibles a los no programadores.

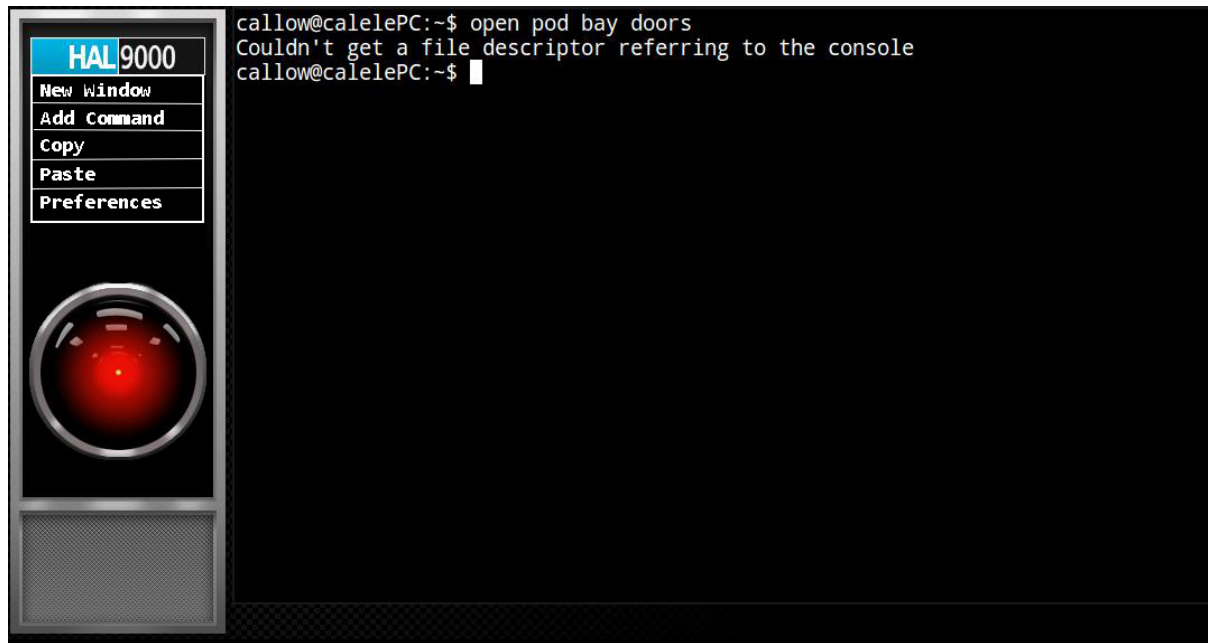
Requerimientos

1. Debe cargar el shell por defecto al iniciar la aplicación, con el usuario actual como el usuario del shell.
2. Debe presentar el gráfico del HAL 9000 al iniciar la aplicación.
3. Debe iniciarse el intérprete de PocketSphinx al iniciar la aplicación, validando que se tiene acceso al micrófono. Si no se logra, debe presentarse un mensaje de error al usuario.
4. La luz de HAL debe tornarse rojo vivo cuando está escuchando los comandos del usuario, verde cuando los está interpretando, y rojo tenue cuando está en modo de espera.
5. Las palabras que habla el usuario deben de ser convertidas en texto que aparece en el shell o en combinaciones de caracteres que recibe el programa que se está ejecutando en el shell.
6. Los comandos que reconoce el programa deben almacenarse en un archivo de texto que contenga un mapeo entre palabras clave y su texto o combinación de caracteres correspondiente.
7. Al iniciar se debe leer el archivo de comandos y utilizar los comandos en el tiempo de la ejecución.
8. Al hablar, PocketSphinx debe convertir el audio en texto y pasar el texto a la aplicación.
9. El texto recibido debe leerse en busca de comandos. Las palabras claves de comandos se reemplazan por su valor si son texto, o se presionan las teclas correspondientes si son combinaciones de teclas. Si no corresponde a comando, se escribe a la terminal literalmente.
10. Para escribir una palabra clave sin que se interprete como comando se debe preceder con la palabra clave "literal". Esto debe imprimir la palabra a la terminal sin ser interpretada. Hablar "literal literal" debe imprimir literalmente "literal".
11. Decir "undo" debe borrar de la terminal los últimos caracteres escritos. Sin embargo no se pueden deshacer las combinaciones de teclas.

12. Decir un número antes de una palabra clave significará realizar esa acción esa cantidad de veces. Por ejemplo "four up" enviará la tecla de arriba cuatro veces.
13. Decir "begin record [keyword]" comenzará a grabar la secuencia de comandos vinculándolos a la palabra clave. "end record [keyword]" finaliza.
14. Posibilidad de elegir entre distintos dispositivos de grabación
15. Cambiar el aspecto visual de la terminal
16. Ver la lista de comandos y poder editar estos en la tabla de la ventana de Commands

Prototipo de interfaz de usuario

Ventana principal:



Ventana de configuración, con sus pestañas respectivas:

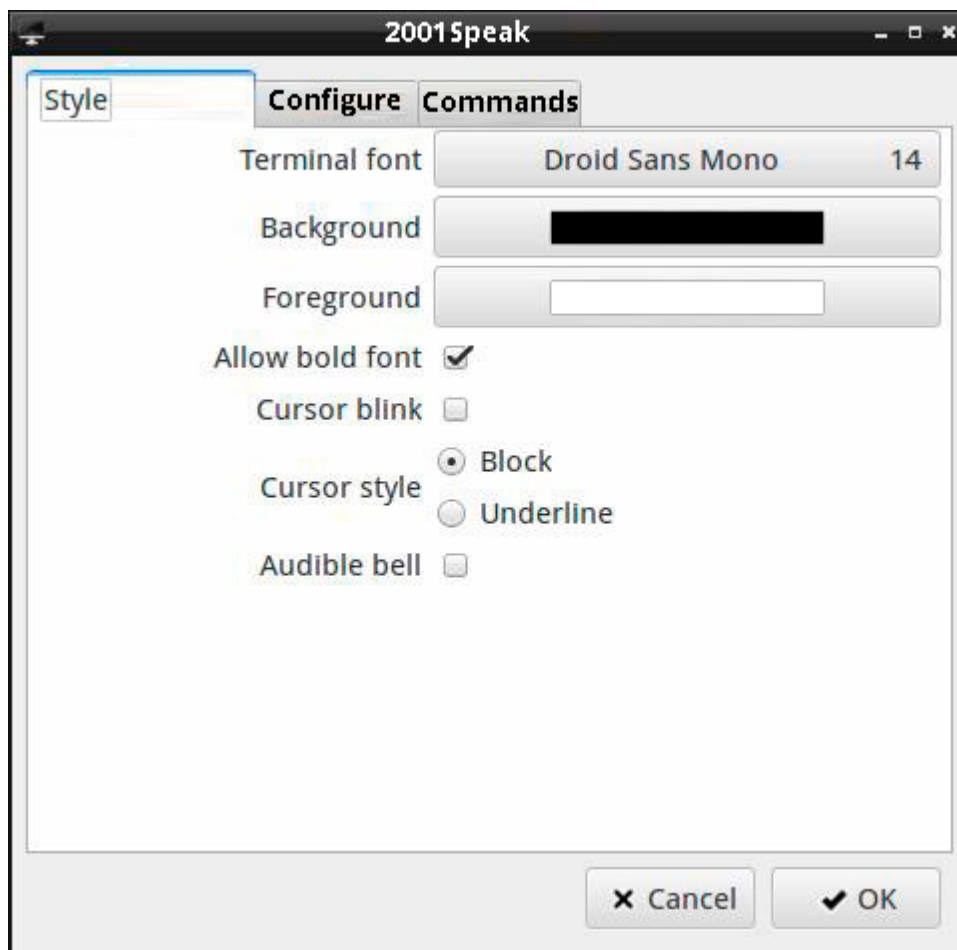
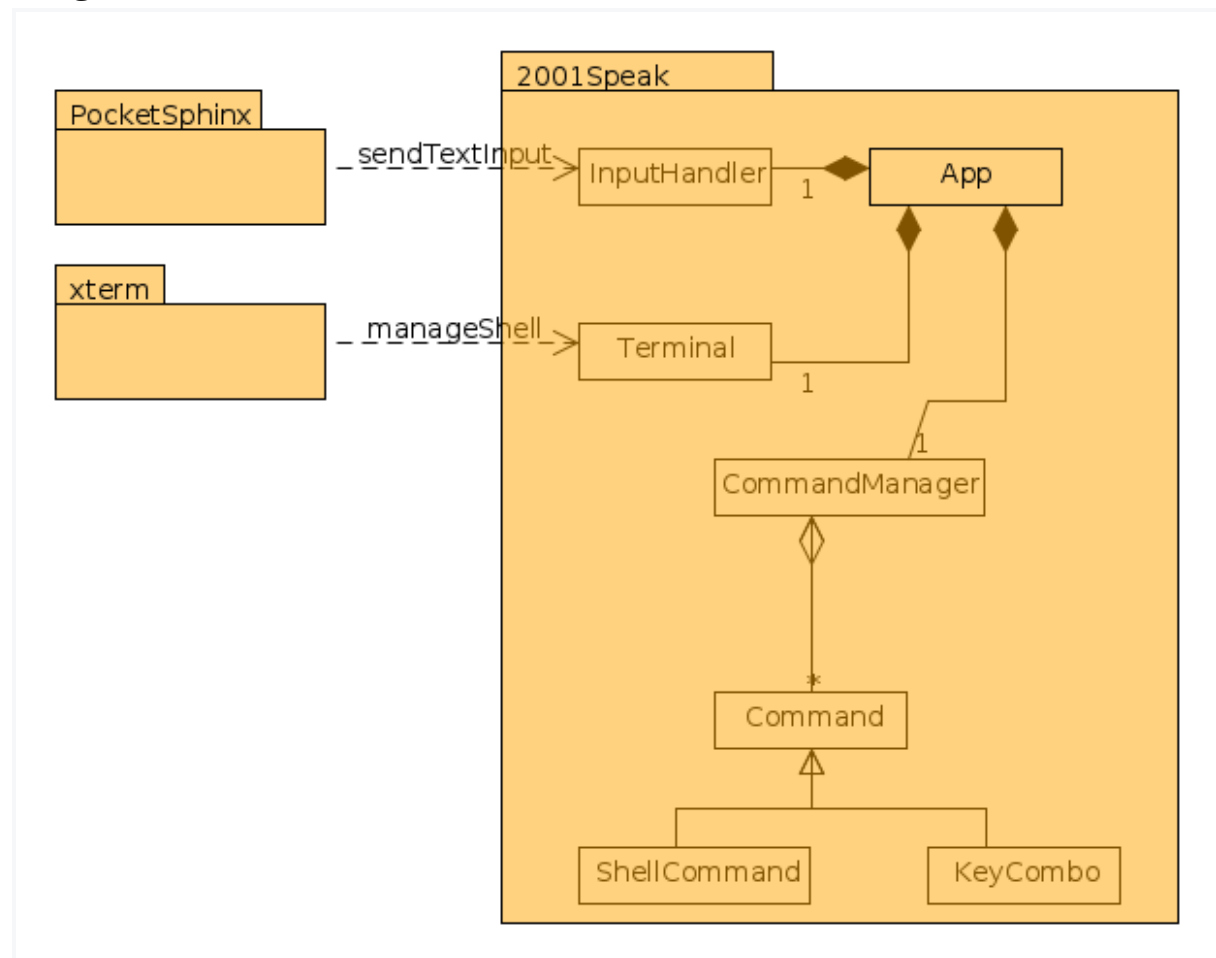


Diagrama de clases



4. Lista de requerimientos lo más completa posible. Debe ser un listado de todas las funcionalidades o consideraciones que hay que tomar en cuenta para la elaboración del sistema.