# PEBL: PA-M.

## Tareas de Percepción Auditiva, Memorama gráfico y Memorama Auditivo.

## Manual de Mantenimiento.

Este documento ilustra los diferentes componentes internos del software PA-M con el fin de proveer una guía para su mantenimiento bajo el modelo de software libre y según la licencia GPL- V3 bajo la cual fue liberado.

Los ejercicios aquí descritos fueron desarrollados según el paradigma propuesto por P. Tallal y M. Piercy:

Tallal, P., & Piercy, M. (1973). Defects of non-verbal auditory perception in children with developmental aphasia. Nature, 241(5390), 468-469.

Tabla de contenido

[PEBL: PA-M. 1](#_Toc486857471)

[Tareas de Percepción Auditiva, Memorama gráfico y Memorama Auditivo. 1](#_Toc486857472)

[Manual de Mantenimiento. 1](#_Toc486857473)

[Archivos Fuentes: 1](#_Toc486857474)

[Lenguaje PEBL: 2](#_Toc486857475)

[Menús: 2](#_Toc486857476)

[Dibujo: 2](#_Toc486857477)

[Sonidos: 2](#_Toc486857478)

[Estructura de funcionamiento: 3](#_Toc486857479)

[Referencia Funciones de PEBL utilizadas en PEBL\_PA-M: 5](#_Toc486857480)

### Archivos Fuentes:

Los archivos fuentes están alojados en Github en la siguiente dirección:

https://github.com/muddokon/PEBL\_PA-M

Los archivos fuentes son documentos de texto plano con codificación UTF-8 y extensión ‘pbl’, se recomienda el uso de un editor de texto para código como ATOM o SUBLIME TEXT.

### Lenguaje PEBL:

El lenguaje PEBL está basado en C. Sin embargo no es fuertmente tipado y no utiliza apuntadores. Una referencia completa a este lenguaje se encuentra en el manual de PEBL.

pebl.sourceforge.net/peblmanual.pdf

### Menús:

Los menús se construyen así:

aviso0 <- MakeLabel("Seleccionar el punto de arranque:",fontWhite)

AddObject(aviso0, ventana)

Move(aviso0, gVideoWidth/2, 100)

La función MakeLabel (string,font) crea una etiqueta de texto sin agregarla a la ventana.

La función AddObject(object,window) agrega cualquier objeto a la ventana.

La función Move(object,x,y) ubica el objeto agregado en la ventana según las coordenadas ‘x’ y ‘y’.

### Dibujo:

El dibujo de elementos gráficos se realiza de la siguiente manera:

La función Draw() refresca los elementos en pantalla, cada llamado produce una actualización de los elementos agregados y eliminados.

flecha1 <- MakeImage("up.png")

La función MakeImage(string) recibe la ruta de una imagen y crea un objeto imagen sin agregarlo a la ventana, es necesario un llamado a la función AddObject para agregarlo y a la función Move para ubicarlo.

RemoveObject(aviso0,ventana)

La función RemoveObject(object,window) elimina un objeto de una ventana.

### Sonidos:

Los sonidos se importan creando un objeto con la ruta al archivo de audio en formato WAV-PCM y almacenando el objeto en una varaiable:

sonidoT1 <- LoadSound("bajo.wav")

La reproducción de los sonidos se realiza mediante las funciones PlayForeground y PlayBackground, al terminar la duración del archivo de audio se detiene. De lo contrario es posible detener la repsoducción por medio de la función Stop().

PlayForeground(sonidoT1)

Wait(duracion)

Stop(sonidoT1)

### Estructura de funcionamiento:

A continuación se describen las mecánicas principales del aplicativo en cuanto a su funcionamiento relacionado con el código en lenguaje PEBL:

#### Lectura de parámetros:

Los parámetros en PEBL son leídos del fichero de configuración que contiene el esquema. Este fichero tiene extensión pbl.schema y posee la siguiente estructura:

duracion|75|La duracion de cada tono en milisegundos

Se trata de tres argumentos separados por caracteres tubería (pipe) donde el primero es el nombre, el segundo el valor default y el tercero la descripción.

En el programa, se llama la función CreateParameters(file,string), se almacena en una variable y se le entrega el nombre del fichero de parámetros:

#parámetros

gParams <- CreateParameters(parametros, "parametros.par")

Finalmente se almacena en varlables separadas cada parámetro descrito para acceder a sus propiedades:

#Variables

duracion <- gParams.duracion

intervalo <- gParams.intervalo

#### Presentación de estímulos:

En primer lugar se declara un arreglo con posiciones según la cantidad de estímulos y la diferenciación de los mismos. Utilizando la función shuffle(array) se recrea el arreglo con posiciones aleatorias para sus elementos.

estimulo <- Shuffle([1,2,3,4])

Cada estímulo es representado junto a el número de la repetición a una realimentación si hay lugar por medio del color de la Fuente donde el color verde representa una retroalimentación positiva y el color rojo una retroalimentación negativa.

SetText(aviso,"Repetición Número: " + c)

SetFont(resultado, fontGreen)

SetText(resultado," ")

Draw()

Dependiendo del estímulo seleccionado en la iteración actual se reproducen los sonidos en orden con el intervalo de separación y la duración indicada en los parámetros. Si se estaba dando retroalimentación se agrega objetos gráficos y se remueven al finalizar la iteración como presentación del estímulo.

if(i==1){

SetText(ayuda,"Iguales")

Draw()

PlayForeground(sonidoT1)

Wait(duracion)

Stop(sonidoT1)

Wait(intervalo)

PlayForeground(sonidoT1)

Wait(duracion)

Stop(sonidoT1)

AddObject(mano2,ventana)

Move(mano2,gVideoWidth/3,alturaManos)

Draw()

Wait(1000)

RemoveObject(mano2,ventana)

Draw()

}

#### Recepción de respuestas:

Las respuestas son recibidas mediante la función WaitForListKeyPress(string[]) que recibe un arreglo de texto con los nombres de las teclas deseadas:

time1 <- GetTime()

resp <- WaitForListKeyPress(["<up>","<down>","<esc>"])

time2 <- GetTime()

#### Recopilación de datos:

La recopilación de datos Inicia con la creación de un archivo de registro con el nombre deseado según el ejercicio que se está practicando.

seguido a esto se escribe la primera línea con los encabezados de las columnas del archivo de texto con separación por comas.

#Crear un archivo de registro para el paciente

gFileOut <- FileOpenAppend(gSubNum+"\_tonos\_ID\_noret.csv")

FilePrint(gFileOut,"repeticion,intervalo,correcto,tiempo")

Para cada iteración se escribe el resultado utilizando las variables de la iteración contador intervalo y la medición temporal como la resta de los dos temporizadores ubicados en el espacio de respuesta.

FilePrint(gFileOut,c+","+intervalo+",si,"+(time2-time1))

Si el experimento es abortado se escribe todas las columnas con la palabra abortado para indicar la interrupción del mismo en el archivo de texto con separación por comas.

FilePrint(gFileOut,"ABORTADO,ABORTADO,ABORTADO,ABORTADO")

### Referencia Funciones de PEBL utilizadas en PEBL\_PA-M:

|  |  |
| --- | --- |
| Función | Utilidad |
| AddObject() | Agregaun objeto gráfico a la ventana como un label. |
| Draw() | Dibuja. Debe ser ejecutadopara refrescar elementos nuevos. |
| EasyLabel() | Crea una etiqueta fácil consólo la ubicación. |
| FileOpenAppend() | Abre o crea un archivo detexto en modo escritura. |
| FilePrint() | Escribe en el archivo detexto una línea. |
| GetTime() | Obtiene el tiempo actual dela máquina. |
| LoadSound() | Carga un sonido. |
| MakeColor() | Crea un color nuevo. |
| MakeFont() | Crea una fuente detipografía para la pantalla. |
| MakeImage() | Carga una imagen en formatoPNG, JPG o BMP. |
| MakeLabel() | Dibuja un cuadro de texto oetiqueta. |
| MakeWindow() | Configura una ventananueva. |
| Move() | Mueve un objeto gráfico enla ventana. |
| PlayBackground() | Reproduce un sonido o video en segundo plano. |
| PlayForeground() | Reproduce un sonido o video en primer plano. |
| PopupMessageBox() | Dibuja un Popup o ventan flotante dentro de la ventana. |
| Print() | Imprime un mensaje en texto en la consola de PEBL. |
| RemoveObject() | Remueve un objeto gráfico de la ventana como un label. |
| SetFont() | Establece la fuente según tipografía, tamaño y color. |
| SetText() | Establece el texto en un elemento gráfico. |
| Shuffle() | Baraja un arreglo de datos. |
| Start() | Función principal que inicia el programa en PEBL. |
| Stop() | Detiene la reproducción de sonido o video. |
| Wait() | Detiene el programa una cantidad de tiempo en milisegundos. |
| WaitForListKeyPress() | Espera para la entrada de teclado con teclas específicas. |