

21721 - Lenguajes de programación

Grupo 01 - 2018/2019



Universitat
de les Illes Balears

Práctica no presencial Lisp

Palabras ocultas

Miguel Vidal Coll - 43225456S - vidalcollmiguel@gmail.com
Antonio Rotger Lopez - 43224332H - tonirotger7@gmail.com

Índice

Introducción	2
Funciones	2
Funciones principales	2
Inicio	2
Mainloop	2
IntroducirLetra	3
Final	3
Final2	4
Funciones de soporte	4
Reset	4
IntroducirNuevo	4
Escribir	4
Funciones de dibujo gráfico	4
PintarFondo	4
DibujarTitulo	5
Paint	5
Cuadricula	5
Funciones de dibujo de imágenes	5
Pintarpalabra	5
Printsec	5
PintarCaracter	5
Dibujarbn	5
Dibujarcolor	6
Dibujar, trozos y reorganizar	6
Programa conversor de .bmp a .img	6
Observaciones	6
Ejemplo de funcionamiento	7

Introducción

En esta práctica se pide realizar el desarrollo de una aplicación, utilizando el lenguaje de programación Lisp, que simule el juego de las palabras ocultas.

Este juego consiste en ir revelando las letras de la palabra que se deben adivinar con ayuda de fragmentos de una imagen relacionada con esa palabra. El jugador irá introduciendo, letra a letra, hasta que adivine la palabra o hasta que agote todos los intentos, en este caso tendrá 17 intentos. El programa también penalizará letras que haya repetido e incluso letras que ya haya descubierto pero que el jugador haya vuelto a introducir.

Cada letra fallida revelará, de forma aleatoria, una 1/16 parte de la imagen correspondiente a la palabra a descubrir. Una vez se haya revelado toda la imagen, al jugador le quedará un intento de deducción de la palabra. Cuando lo agote, se habrá acabado la actual jugada. En los dos casos (acertar o fallar la palabra) se informará al jugador de que ha acabado la jugada y podrá visualizar la imagen completa, su puntuación total y si desea continuar.

El juego tiene un sistema de puntuación donde el jugador empieza con 255 puntos y por cada letra fallada se le restarán 15 puntos. Por cada nueva jugada se le sumarán 255 puntos. Finalmente esta puntuación se guardará junto al nombre del jugador.

Funciones

Funciones principales

Inicio

Primera función del programa que, como dice el nombre, inicia la ejecución del mismo. En esta se pinta el primer menú de interacción con el usuario para que introduzca el nombre con el que va a jugar.

Para pintar el menú llama a la función *"pintarfondo"* la cual pinta el fondo de azul y a continuación dibuja un rectángulo con las herramientas de pintado de lisp donde dentro de este se introducirá el nombre.

Mainloop

Función principal donde en primer lugar prepara las variables necesarias con la función *"reset"* y a continuación leerá una palabra del fichero llamado "palabras.txt" según el nº de jugada en la que esté. Si no es el final del fichero, es decir, si aun quedan palabras por jugar, el programa continúa, sino acabará llamando a la función *"final2"*.

Si ha leído una palabra correctamente dibujará la interfaz y llamará a la función “*introducirLetra*” pasando por parámetro los puntos actuales, el nº de jugada, el nombre del jugador y el nº de intentos.

Esta función se ejecutará una vez por jugada por tanto es donde se prepara la partida para una jugada, iniciando los puntos, jugadas e intentos, a parte de la palabra a adivinar.

IntroducirLetra

Esta función es la que lleva el funcionamiento de una jugada. Se introducen las letras para adivinar la palabra, dibuja los fragmentos de la imagen y las letras y va actualizando la puntuación.

En primer lugar se pide la introducción de una letra que será evaluada por el programa comparando con una lista donde estarán todas las letras introducidas por el momento en la jugada. habrá dos condiciones:

- La letra está dentro de la lista: querrá decir que ya se ha introducido previamente y ya se ha analizado, por tanto será tratada como letra fallada independientemente de si es una letra de la palabra a adivinar ya que sí está en la lista querrá decir que esa letra ya se ha acertado.
- La letra no está en la lista: entonces será introducida dentro de la lista y se analizará si es parte de la palabra a adivinar. Si lo es será dibujada en la interfaz dentro de la parte reservada de la palabra, sino será tratada como fallada.

De esta manera se evita que se cuente por dos si la letra ya ha sido adivinada.

Si la letra ha sido determinada como fallada se dibujará dentro de la interfaz en la parte reservada para las letras falladas y se mirará si aún quedan intentos. Estos intentos van determinados por una variable que va aumentando cada vez que se introduce una palabra. Si quedan intentos se dibujará un fragmento de la imagen y si no quedan intentos saldrá del bucle de introducir letras. El programa también saldrá una vez se haya acertado la palabra la cual cosa se analiza comparando dos integers, uno con la longitud de la palabra y otro con el número de letras acertadas.

Una vez acabado el bucle la función llamará a la función “*final*” pasando por parámetro los puntos totales, el nº de jugadas, la palabra de la última jugada y el nombre del jugador de la partida.

Final

Esta función dibuja el menú de interacción con el usuario entre jugadas. Esta dibuja un fondo azul con la función “*pintarfondo*” y en él dibuja la imagen con la que se jugaba en el momento, la palabra de la imagen, la puntuación total conseguida acumulada y se pregunta al jugador si desea hacer otra jugada. Puede haber dos situaciones:

- El jugador introduce "S": Se llama a la función "*mainloop*" pasándole por parámetro el nombre del jugador, los puntos conseguidos y el número de jugada para leer la siguiente palabra y preparar la siguiente jugada.
- El jugador introduce "N": Se llama a la función "*escribir*" pasando por parámetro los puntos y el nombre del jugador donde se escribirán en un fichero llamado "puntuacion.txt" y acabará el programa.

Final2

Esta función es similar a la función "*final*" pero solo será llamada cuando se hayan terminado las palabras del fichero "palabras.txt" (se haya llegado al final del fichero). Ésta dibujará un fondo azul y un rectángulo y dentro escribirá la puntuación total y un mensaje haciendo saber al jugador de que no hay más palabras.

Finalmente llama a la función "*escribir*" pasando por parámetro los puntos y el nombre del jugador donde se escribirán en un fichero llamado "puntuacion.txt" y acabará el programa.

Funciones de soporte

Reset

Esta función reiniciará los valores de las variables que se utilizan para la fragmentación de la imagen y que será necesario para poder hacer una nueva jugada en el programa.

IntroducirNuevo

Esta función compara los valores de una lista con una letra (o valor) introducida para poder saber si ya ha sido introducida previamente.

Escribir

Función para escribir el resultado final en un fichero llamado "puntuación.txt". Si el fichero ya existe, escribe en él y sino, lo crea y escribirá en él. Por parámetro tiene las variables puntos y jugador que serán los valores que le entrarán y escribirá en el fichero.

Funciones de dibujo gráfico

PintarFondo

Función para dibujar el fondo del menú de color azul.

DibujarTitulo

Función para dibujar las letras de la interfaz que no serán modificadas como son las del título y las de las secciones de jugador, puntos y jugadas. Para cada letra se llama a la función “*dibujarbn*” con la letra a dibujar, la posición donde se deba dibujar y el tamaño de la imagen de la letra.

Paint

Función que dibuja las líneas de la estructura de la interfaz como son los separadores entre secciones y los marcos de las letras. Para cada sección de la interfaz que necesita una línea de marcos, se llama a la función “*cuadricula*” donde se pasa por parámetro la posición, el número de marcos necesarios y el tamaño del marco.

Cuadricula

Función para dibujar los marcos de las letras de la interfaz. Le entra por parámetro la posición, el número de marcos a dibujar y el tamaño, entonces se posiciona y dibuja un marco según el tamaño. Lo repite tantas veces como se haya indicado.

Funciones de dibujo de imágenes

Pintarpalabra

Función para dibujar las letras en la interfaz de la palabra a adivinar. Le entra por parámetro la letra a dibujar y la posición donde se debe dibujar de la palabra. Entonces busca el archivo, se posiciona y llama a la función “*dibujarbn*”.

Printsec

Función para dibujar en la interfaz una palabra carácter a carácter como puede ser “jugador” o “puntuacion”. Le entra por parámetro la palabra y la posición donde se debe dibujar. Entonces guarda la longitud de ésta y va dibujando cada letra llamando a la función “*pintarCaracter*”

PintarCaracter

Función para pintar un carácter en la interfaz. Le entra por parámetro el carácter en cuestión, la posición y el tamaño, entonces busca el archivo, se posiciona y llama a la función “*dibujarbn*”.

Dibujarbn

Función para dibujar imágenes en blanco y negro en formato “.img”.

Dibujarcolor

Función para dibujar imágenes en color en formato “.img”.

Dibujar, trozos y reorganizar

Funciones para dibujar fragmentos, en la interfaz, de una imagen en color en formato “.img” sin repetir los fragmentos.

La función “*dibujar*” dibuja el fragmento según los límites de píxeles que se han establecido en la función “*trozos*” y en “*reorganizar*” se evita la repetición de fragmentos.

Programa conversor de .bmp a .img

El programa usado para convertir las imágenes en formato “.bmp” a “.img” ha sido realizado con lenguaje Java.

Dentro de la carpeta raíz del programa encontramos cuatro carpetas, dos de entrada y dos de salida, en la que se guardan las imágenes. La razón por la que tenemos dos carpetas para cada tipo de archivos es debido a que las imágenes en color se tratan de forma diferente a las imágenes en blanco y negro, además de conseguir encontrar fácilmente las imágenes a color entre todas las letras proporcionadas por el profesor.

El programa cuenta con tres arrays: uno con los nombres de todas las letras y números, otro de los distintos tipos de imagen en blanco y negro, y un tercero con los nombres de las imágenes a adivinar. Combinando los dos primeros conseguiremos todos los archivos a convertir en blanco y negro.

El método para convertir imágenes en blanco y negro crea un array de tres posiciones (BGR), saltará los 54 primeros bits de la cabecera que no aportan información sobre los colores de la imagen, y irá escribiendo en el archivo de salida un 0 o un 1 dependiendo del valor del bit de color rojo, ya que como se ha explicado en clase, este es el más significativo para saber si se debe pintar la imagen de blanco o de negro.

En caso del conversor de imágenes a color, debemos saltar los primeros 138 bits, ya que le conversor que hemos utilizado para convertir las imágenes JPEG a BMP añade bits innecesarios para nuestro propósito. Podemos saber cuántos observando las propiedades de la imagen y viendo en cuantos bits difiere respecto a los que debería tener. Una vez saltada la cabecera, escribimos en el nuevo archivo cada uno de los valores de los bits BGR.

Observaciones

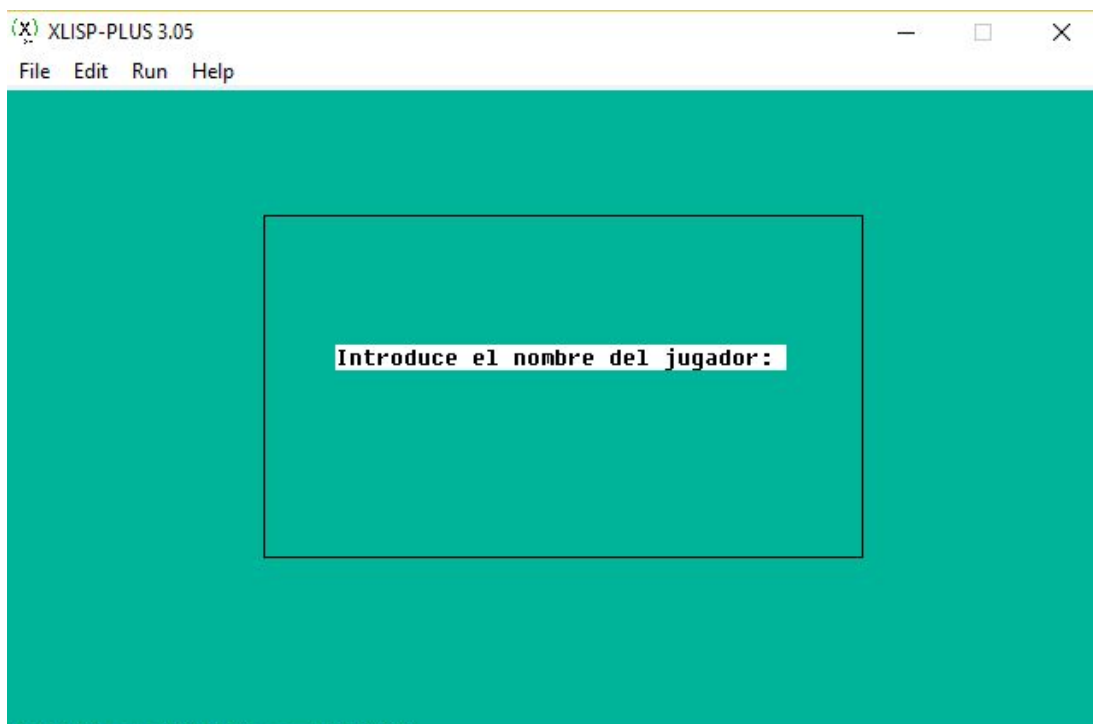
Las palabras en el fichero de “palabras.txt” deben coincidir con el nombre de las imágenes en “.img” que se vana a usar en el programa ya que las palabras que se leen en el fichero son usadas para la búsqueda de la respectiva imagen.

El programa es entregado con unas determinadas imágenes en formato “.img” con las que se ejecutará el programa. Éstas están dentro de la carpeta “fotosIMG”. Para leer bien la palabra y la imagen en el programa, la palabra del fichero “palabras.txt” debe corresponder con el nombre de la imagen que se quiere visualizar con esa palabra. Si se desean introducir nuevas imágenes, deberán guardarse dentro de la carpeta “fotosIMG” y modificar el fichero “palabras.txt” de manera correcta, introduciendo la palabra correctamente como se ha dicho previamente.

Dentro de la carpeta “IMG” se encuentran las imágenes para las letras.

Ejemplo de funcionamiento

En primer lugar cargamos dentro del intérprete Lisp el archivo con (load “Practica1.lsp”). Entonces directamente aparecerá el primer menú de la siguiente manera:



Introducimos el nombre con el que vamos a jugar y comenzará el juego.

XLISP-PLUS 3.05

File Edit Run Help

P	A	L	A	B	R	A	S									
O	C	U	L	T	A	S										
J	U	G	A	D	O	R							M	T	V	R
P	U	N	T	O	S									2	5	5
J	U	G	A	D	A	S										1

--	--	--	--	--	--	--	--

LETRAS FALLADAS --->

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

introduce una letra: _

Cuando introducimos una letra habrá dos posibilidades, que hayamos acertado:

XLISP-PLUS 3.05

File Edit Run Help

P	A	L	A	B	R	A	S									
O	C	U	L	T	A	S										
J	U	G	A	D	O	R							M	T	V	R
P	U	N	T	O	S									2	5	5
J	U	G	A	D	A	S										1

A							
---	--	--	--	--	--	--	--

LETRAS FALLADAS --->

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

introduce una letra: a

o que hayamos fallado:

XLISP-PLUS 3.05

File Edit Run Help

P	A	L	A	B	R	A	S
O	C	U	L	T	A	S	
J	U	G	A	D	O	R	
M	T	V	R				
P	U	N	T	O	S		
2	5	5					
J	U	G	A	D	A	S	
							1

--	--	--	--	--	--	--	--


LETRAS FALLADAS ---> Q

introduce una letra: g

Una vez hayamos agotado todos los intentos o hayamos acertado la palabra saldrá el segundo menú para seguir con la partida con una nueva jugada o finalizar:

XLISP-PLUS 3.05

File Edit Run Help

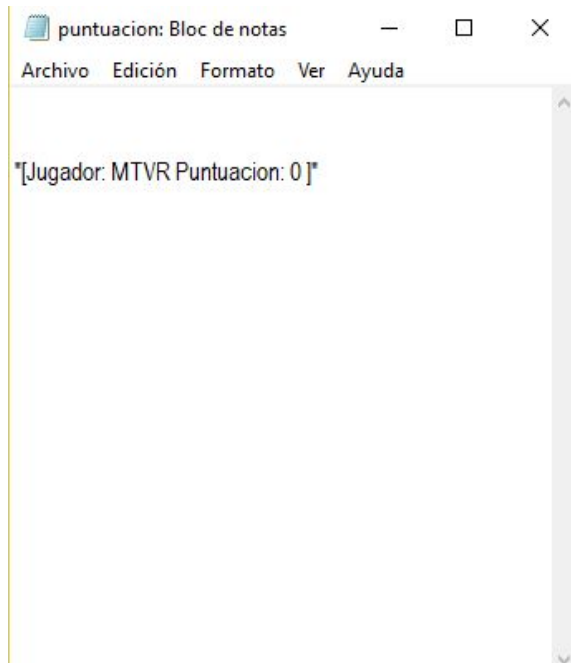


"ajedrez"

Puntuacion: 0

Quieres volver a jugar? S/N

Si finalizamos se escribirá el resultado en el fichero "puntuación.txt":



Si deseamos continuar aparecerá la siguiente palabra con la misma interfaz que la segunda imagen y será el mismo procedimiento.

Existe otro caso que aparecerá una vez se hayan jugado todas las palabras posibles que hará aparecer la siguiente interfaz:



Finalmente se acabará el programa escribiendo el resultado en el fichero mencionado anteriormente.