INFORME DE

INTRODUCCION A LA PROGRAMACION

Trabajo Práctico

“Pasa Palabra”

Comisión 2

Docentes: Veronesi Luis, Bottino Flavia, Benesch Alicia.

Alumnos: Manuel Acosta, Cáceres Marcos

El trabajo practico consiste en editar un juego llamado “Pasa palabra” (prediseñado por consola en Python)

Este juego consta de un límite de 60 segundos en los cuales el usuario puede adivinar las palabras escondidas que le alcancen, donde cada error quita puntos pero también se podrá pasar de palabra sin perder puntuación (función agregada por nosotros). Las palabras son las más largas extraídas de los subtítulos de la película “Matrix” con algunas de sus letras reemplazadas por asteriscos.

Al ejecutar el programa principal se abrirá una ventana con un fondo de inicio dando una bienvenida al juego, una breve explicación de la mecánica de juego, cómo funciona el sistema de puntos y un mensaje para pasar de pantalla y poder empezar a jugar. (Este fondo lo editamos con fuentes personalizadas y opciones de fusión en Adobe Photoshop a nuestro criterio)

En la siguiente pantalla, con otro fondo y otro mensaje (que nos dice como pasar de palabra), se muestran las palabras extraídas ya modificadas por asteriscos donde el usuario debe adivinar dicha palabra, ésta se evaluará con dos tipos de aciertos y un error:

* Acierto: es cuando la palabra ingresada por el usuario cumple con las letras que se muestran en pantalla pero no es la palabra que está oculta, sumará puntos equivalentes a la longitud de la palabra ingresada por el usuario. \*sonará sonido “acierto”\*
* Palabra Exacta: es cuando la palabra ingresada por el usuario es exactamente la misma que está ocultando, multiplicará 5 veces los puntos equivalentes a la longitud de la palabra escondida. \*sonará sonido “correcto”\*
* Error: Es cuando la palabra es incorrecta, se restaran los puntos equivalentes a la longitud de la palabra escondida. \*sonará sonido “error”\*

En ésta pantalla también decidimos mostrar la palabra escondida sin ningún asterisco en caso de que sea erróneo el ingreso y un mensaje de incentivación a seguir acertando en caso de una seguidilla de aciertos(\*sonará sonido de “seguidilla”\*).

Una vez terminado el tiempo o terminada el listado de palabras se reproducirá un sonido de fin de juego pasando al fondo de fin de juego, donde se muestra el puntaje obtenido y algunas reseña personalizadas para dejar más presentable el final del juego.

Con respecto al diseño de sonidos optamos por una variedad de sonidos personalizados de distintas fuentes.

Las imágenes fueron descargadas de internet y editadas con Adobe Photoshop por nosotros.

Las explicaciones de las funciones y dichos cambios realizados en el programa principal se encuentran comentadas en dicho programa paso por paso, así como también otras pruebas que tuvimos que hacer, lo cual nos sirvió mucho para poder jugar más con el programa y entender ciertos parámetros que fueron necesarios para la personalización a nuestro estilo del juego.

De todas formas a continuación dejamos las funciones copiadas tal cual están en “FuncionesVacias”:

def ocultar(cadena): #reemplaza caracteres seleccionados en la variable "vocales" por \*\*\*\*\*

caden=""

vocales="aeu"

for caracter in cadena:

if caracter in vocales:

caden=caden+"\*"

else:

caden=caden+caracter

return caden

def norepetir\_palabra(x,y): #funcion auxiliar para la funcion nuevapalabra

if x in y:

return True

else:

return False

def elegirLaMasLarga(listapalabras,listaocupada): #elige la polabra mas larga de la listapalabrasPeli y se fija que no se haya usado

largo=len(listapalabras[0])

palabra=listapalabras[0]

for i in range(len(listapalabras)):

if len(listapalabras[i])>largo and listapalabras[i] not in listaocupada:

largo=len(listapalabras[i])

palabra=listapalabras[i]

listaocupada.append(palabra)

return palabra

def damePalabras(frases): #funcion auxiliar para frasesToPalabras

cadenas=[]

palabra=""

cont=0

seAgrego=True

entro=False

for elemento in frases:

seAgrego=False

if elemento==" " :

cadenas.append(palabra)

palabra=""

seAgrego==True

elif elemento!=" ":

palabra=palabra+elemento

cont=+1

if palabra not in cadenas:

cadenas.append(palabra)

return cadenas

def frasesToPalabras(lista): #esta funcion saca palabras de una frase y las mete en una lista llamada cadenas con la ayuda de la funcion damePalabras

cadenas=[]

palabra=""

aux=""

cont=0

for i in range(len(lista)):

cadenas=cadenas+damePalabras(lista[i])

return cadenas

def quitarCaracteresEspeciales(linea): #quita los caracteres especiales de una cadena ya antes tomada

linea=linea.lower() #hacer todas minus

caracespeciales="àèìòùáéíóú"

a=0

vocales="aeiouaeiou"

signos="!¿?¡.,-\_ç´`+¡;:'0987654321ºª\*/+<>"

nuevapal=""

for carac in linea:

if carac in caracespeciales:

for i in range(len(caracespeciales)):

if caracespeciales[i]==carac:

a=i

nuevapal=nuevapal+vocales[a]

elif carac not in signos:

nuevapal=nuevapal+carac

return nuevapal

def lectura( listaPalabrasPeli): #lee el archivo de subtitulos y lo agrega a la listapalabrasPeli

palabras1=open("Matrix1999.txt","r")

a=palabras1.readlines()

for elemento in a:

listaPalabrasPeli.append(elemento[0:-1])

palabras1.close()

def lectura2(listaPalabrasDiccionario): #lee el archivo lemario y agrega a la listaPalabrasDiccionario

diccionario=open("lemario.txt","r")

b=diccionario.readlines()

for elemento in b:

listaPalabrasDiccionario.append(elemento[0:-1])

diccionario.close()

def seleccionDeLetras(palabra): #selecciona las letras de la palabra

nuevapal=""

for letra in palabra:

nuevapal=nuevapal+letra

ocultar(nuevapal)

return nuevapal

def nuevaPalabra(listapalabras,palabras\_ocupadas): #toma una nueva palabra al azar de la lista de palabras y se fija que no se repita con la funcion no repetir

palabra="" #cadena vacia

num=random.randint(0,len(listapalabras)-1) #asigna un valor al azar en el rango de la lista

palabra=listapalabras[num] #hace el sorteo

repite=norepetir\_palabra(palabra,palabras\_ocupadas) #llama a no repetir\*

while repite==True: #ciclo siempre que se repita va a volver a elegir un numero al azar! Je

num=random.randint(0,len(listapalabras)-1)

palabra=listapalabras[num]

repite=norepetir\_palabra(palabra,palabras\_ocupadas)

palabras\_ocupadas.append(palabra) #lo mete en una lista

## listapalabras.append(palabra)#prueba

return palabra

def esValida(candidata, oculta, palabra): #devuelve 1 si la candidata es igual a la palabra, 2 si el len de candidata es igual a el len de oculta or al len de palabra

if candidata==palabra: #y si es erronea -1

return 1

elif len(candidata)==len(oculta) or len(candidata)==len(palabra):

return 2

else:

return -1

def puntos(listapalabras,palabra,listaPalabrasDiccionario): #puntuacion que se usa en la funcion procesar

puntos=0

if palabra in listapalabras:

puntos+=len(palabra)\*5

elif palabra in listaPalabrasDiccionario:

puntos+=len(palabra)

return puntos

def procesar(candidata, oculta, palabra, listapalabras, listaPalabrasDiccionario): #realiza todo el proceso de las palabras y el puntaje con la funcion puntos

return puntos(listapalabras,palabra,listaPalabrasDiccionario)

def dibujoerrada(screen,errada): #agregue esta funcion para poner la palabra correcta cuando el usuario erra pero decidimos dejarla para caso de prueba

#ya que encontramos una solucion que ahorra codigo

defaultFont= pygame.font.Font( pygame.font.get\_default\_font(), TAMANNO\_LETRA)

defaultFontGrande= pygame.font.Font( pygame.font.get\_default\_font(), TAMANNO\_LETRA\_GRANDE)

#dibuja la errada

screen.blit(defaultFont.render(errada, 1, COLOR\_TEXTO), (160, 470)

Alumnos: Acosta, Manuel - Cáceres, Marcos