

A thick dark blue vertical bar runs down the left side of the page. A blue arrow-shaped box points to the right from this bar, containing the delivery date. Below the bar, several thin, curved lines in dark blue and light grey sweep upwards and to the right, creating an abstract graphic element.

Fecha de entrega: 16 de
Octubre del 2016

Documentación Externa

Proyecto programado Master Mind

Estudiante Danny Xie Li - Carnet 2016086098
Profesor: William Mata
INSTITUTO TECNOLÓGICO DE COSTA RICA
ESCUELA DE COMPUTACIÓN
CURSO TALLER DE PROGRAMACIÓN

Contenidos

Enunciado del proyecto.....	2
Temas de investigados.....	5
Conclusiones.....	15
Estadística de tiempo.....	16
Rubrica de evaluación.....	17
Anexos.....	18

1. Enunciado del proyecto

Master Mind es un juego de mesa que requiere lógica y memoria. Participan dos jugadores: un jugador hace una combinación de elementos, por lo general son colores, y el otro jugador trata de determinar los colores de esa combinación y el orden en que fueron puestos. El jugador que hizo la combinación de elementos debe dar información al otro jugador cuando este haga un intento para determinar esa combinación. Hay un área de calificación que existe por cada jugada. Si el jugador que está tratando de determinar la combinación pone un elemento que está en la combinación y en la posición correcta se califica con un color negro, si el elemento está en la combinación, pero no en la posición correcta se califica con un color blanco. Este programa implementará una versión de este juego.

A) JUGAR

Esta opción del menú permite jugar el Master Mind. Cuando se escoge esta opción se muestra una ventana similar a la que está en el manual de Master Mind, en la sección de ¿Cómo jugar? según las condiciones establecidas en la opción de configuración del juego.

B) CONFIGURAR

Esta opción es para indicar las condiciones con que se va a jugar. Están las siguientes opciones: Nivel de dificultad: Fácil (8 jugadas, combinación de 4 elementos de 6), Medio (7 jugadas, combinación de 4 elementos de 6), Difícil (6 jugadas, combinación de 4 elementos de 6). Reloj: Si, No, Cronómetro por jugada, Cronómetro por juego, Posición del panel de elementos: Derecha o izquierda, Panel de elementos para usar en la combinación: Colores, Letras, números, emoticones, Nivel: Fácil, Medio y Difícil.

En caso de seleccionar "Si" el reloj va a ser un cronómetro. En caso de seleccionar "No" entonces no aparece el cronómetro ni se consideran

los juegos para los Top-10. En caso de seleccionar "Cronómetro" debe poner los tiempos máximos que se van a dar ya sea para completar una jugada o el juego completo según haya escogido. Aquí los datos son obligatorios: las horas pueden estar entre 0 y 2, los minutos entre 0 y 59 y los segundos entre 0 y 59.

C) TOP 10 - RESUMEN

Esta opción despliega en una sola pantalla los records de los mejores 10 primeros jugadores por cada nivel de dificultad: aquellos que hicieron menos tiempo para completar el juego. En la opción debe indicarse el nivel que se va a consultar. En caso de no tener los 10 jugadores en algún nivel se despliegan los que se tengan. El Top 10 se guarda en el archivo "mastermind2016top10.dat". El despliegue es resumido: nombre y tiempo total que duró en completar el juego.

D) TOP 10 - DETALLES

Esta opción despliega los records de los mejores 10 primeros jugadores por cada nivel de dificultad: aquellos que hicieron menos tiempo para completar el juego. En la opción debe indicarse el nivel que se va a consultar. En caso de no tener los 10 jugadores en algún nivel se despliegan los que se tengan. El Top 10 se guarda en el archivo "mastermind2016top10.dat". El despliegue es detallado: nombre del jugador, tiempo total que duró en completar el juego, combinación del juego, fecha y hora en que jugó, el tiempo que duró en hacer cada jugada. Note que la suma de los tiempos por cada jugada da el tiempo total.

E) Ayuda

Esta opción la usaremos para que el usuario pueda ver el Manual de Usuario.

F) Acerca de

Esta opción la usaremos para desplegar información “Acerca del programa” donde colocaremos al menos los datos del nombre del programa, la versión, la fecha de creación y el autor.

G) Salir

Esta opción se usa para salir de este programa (también se puede salir con el botón de cerrar “X” en la interfaz gráfica), regresa a la ventana inicial donde está el pseudo -teléfono

2. Temas de investigados

Interfaz Gráfica usada: Librería Tkinter de Python.

Elementos Usados:

- **Tk()** : Esta opción le permite al programador crear una ventana.
¿Cómo se usa?
Cuando se desea crear una ventana llama esta función y se lo asigna una variable (es recomendado) para futuros usos. Por ejemplo:
`Ventana = Tk ()`
- **Variables globales**: Es un tipo de variable libre en la cual se orienta en el ámbito del programa principal.
¿Cómo se usa?
En el programa principal se define la variable global, que debe contener estos 3 elementos: el nombre de la variable, la asignación y el valor. Por ejemplo:
`Lista = []`
Si se quiere usar esta variable en una función se llama esta variable por medio de la palabra global. Por ejemplo:
`global Lista`
- **Toplevel (Ventana)**: Esta opción permite al programador crear una pantalla secundaria a la pantalla principal.
¿Cómo se usa?

Cuando se desea crear una pantalla secundaria al principal se escribe esa función y entre paréntesis la pantalla principal o la pantalla que desea relacionarlo. Por ejemplo:
`VentanaSecundaria = Toplevel (Ventana)`

- **Button (opciones):** Esta opción le permite al programador crear botones y agregarlos a la pantalla, en donde se le puede agregar diferentes opciones como el tamaño de la letra, color del botón, la acción que desea realizar y en que pantalla se debe contener.

¿Cómo se usa?

A la opción Button se le da las opciones entre paréntesis y separados por comas, en donde lo desea ubicar, en que pantalla, color, letra, entre otros, además se le debe empaquetar el botón para que aparezca en la pantalla de lo contrario no existe. Por ejemplo:
`Botón = Button (Ventana, text = "Haga click").pack ()`

- **Label (opciones):** Al igual que el botón, esta opción le permite crear etiquetas de textos o imágenes, posee las mismas opciones que el botón.

¿Cómo se usa?

A la opción de Label se le da unos conjuntos de opciones separados por coma y entre paréntesis todas las opciones que se desea asignar. Además se le debe empaquetar la etiqueta para que aparezca en la pantalla de lo contrario no existe. Por ejemplo:
`Etiqueta = Label (Ventana, text = "Hola").pack ()`

- **Canvas (opciones) y Frame (opciones):** Estas dos opciones le permiten al programador crear espacios dentro de una pantalla principal, en la opción de Canvas () le permite crear objetos, agregar botones, etiquetas entre otros en cambio el Frame todo lo mencionado anteriormente con excepción con crear objetos.

¿Cómo se usa?

`MiniEspacio = Canvas (Ventana, bg = "blue").pack ()`

`MiniEspacio = Frame (Ventana, text = "Hola", fg = "red").pack ()`

En las opciones se deben poner la ventana en la que se quiere insertar y empaquetarlo o ubicarlo en la pantalla según pixeles o filas y columnas.

- **PhotoImage (file = NombreDelArchivo):** En esta opción le permite al programador agregar imágenes a la pantalla por medio de esta función.

¿Cómo se usa?

La función PhotoImage, entre paréntesis se escribe la palabra file, el símbolo de asignar y el nombre del archivo en formato string. Por ejemplo:

```
Imagen = PhotoImage ( file = "hola.gif" )
```

- **Entry (opciones):** Esta función le permite al programador agregar campos de texto a la interfaz gráfica. Además para tomar el valor que está en el campo de texto se usa una variable de tipo string o tipo entero.

¿Cómo se usa?

Para crear un campo de texto se le pone las opciones entre paréntesis, la variable y empacarlo.

```
Variable = StringVar () #Se define la Variable como un valor string.
```

```
Espacio = Entry (Ventana, Variable).pack () #Se le asigna el campo de texto a una variable.
```

```
Variable.get () #Toma el valor que tiene la variable.
```

```
Variable.set ("Adiós") #Pone un valor a la variable.
```

- **tk.ComboBox (opciones):** Esta función crea algo similar a una list box pero no lo es, es un tipo de botón que despliega una tira de valores que le permite al usuario seleccionar un valor de todos los que tiene en esa caja.

¿Cómo se usa?

Se llama la función tk.ComboBox y entre paréntesis las opciones que desea agregar, Además para agregarle valores se le asigna una lista de valores a la variable values. Por ejemplo:

```
Valor = StringVar () #Se le asigna el valor de string a la variable.
```

```
Combo = tk.ComboBox (Ventana, textvariable = valor, values = ["Casa", "Trabajo", "Otros"]).pack () #Se le asigna el combobox a la variable.
```

```
Valor.get () #Para conocer el valor que se escogió.
```


- **messagebox.show Tipo (título,mensaje):** Esta función le permite al programador mandar un cuadro de texto dependiendo del tipo que se quiere mandar puede ser de error, información, advertencia, entre otros.

¿Cómo se usa?

Se llama a la función escribiendo `messagebox.show`, el tipo de cuadro de texto que se quiere mandar y entre paréntesis el título del cuadro en formato string y el mensaje que se quiere transmitir también en formato string. Por ejemplo:

```
messagebox.showerror ("Error", "Debe escribir un nombre")
```

- **Radiobutton (opciones):** En esta función le permite al programador crear botones en donde el usuario puede escoger entre las opciones que existe, su forma es de un círculo (botón) y el enunciado.

¿Cómo se usa?

Se escribe la función `radiobutton` y entre paréntesis las opciones que desea que contenga y además se le debe poner un valor al `radiobutton` y una variable para poder conocer su valor. Por ejemplo:

```
Radiobutton (ventana,text=texto,variable=IntString,value=1).pack()
```

- **os.system (ubicación, nombre del archivo):** Es un módulo que le permite al programador ejecutar archivos, en el programa de contactos fue usado para ejecutar sonidos o música en formato .wav.

¿Cómo se usa?

Se debe llamar la función de `os.system()`, y en formato string la ubicación del archivo y el nombre del archivo. Por ejemplo:

```
os.system ("start E:\BeethovenMoonlight.wav")
```

- **tk.Notebook (opciones):** El propósito de este widget es permitir que el usuario pueda seleccionar ventanitas de contenidos haciendo click en los diferentes pestañas.

¿Cómo se usa?

Se llama el widget y entre paréntesis las opciones que desea agregarle.

```
nbook = tk.Notebook(ventana1).pack() #El widget se le asigna a la variable.
```

```
frame1 = Frame ( ventana1) #Se debe crear una ventanilla
```

```
tab1=nbook.add ( frame1, image = buscar,padding = 10) #Se le  
agrega una pestaña al widget.
```

- **Startfile("Nombre del Archivo"):** Esta opción le permite al programador abrir archivos.
¿Cómo se usa?

Se escribe la función y entre paréntesis y en formato string el nombre del archivo que desea abrir. Por ejemplo:

```
start file ("Manual-De-Usuario-DContact.pdf")
```

- **Scrollbar(opciones):** Esta función le permite crear un scrollbar a la ventana, se le debe poner en las opciones la ventana en donde la desea poner y otras opciones si lo desea.
¿Cómo se usa?

Se pone la función, la ventana en donde lo desea ubicar, en que orientación (vertical o horizontal). Por ejemplo:

```
Barra1 = Scrollbar ( frame2, orient = "vertical", command =  
canvas1.yview)
```

- **Ttk.Scale(opciones):** Esta opción le permite definir una escala de valores y debe poner el rango de valores que desea tener esto.

¿Cómo se usa?

Se escribe la función, y le pone valores que desea, como el comando, los rangos de valores, la ventana, variable, tamaño, entre otras opciones.

```
Ttk.Scale(ventana, from_=0,  
to_=100,length=200,variable=dificultad,command=  
DefinirDificultad).place(x=35,y=70)
```

Librerías usadas

Antes de usar los elementos de la librería se deben importarla, con la palabra import :

- ***import time:*** Esta librería le permite acceder todos los contenidos relacionados con el tiempo.
- ***import random:*** Esta librería permite obtener datos aleatorios.
- ***Import sys o import os:*** Esta librería permite acceder a funcionalidades del sistema operativo, que nos permite manipular la estructura de directorios. *Import startfile:* Permite abrir archivos.
- ***import tkinter.ttk as ttk:*** Esta librería contiene diferentes versiones de los elementos estándares de Tkinter.
- ***from tkinter import *:*** Esta librería nos permite acceder a todos los componentes gráficos que posee Tkinter para desarrollar una interfaz gráfica.
- ***from os import startfile:*** Esta opción le permite acceder todos los contenidos de os, en donde especifica que le permite abrir archivos, en el caso de este trabajo archivos pdf.
- ***from tkinter import messagebox:*** Esta librería le permite al usuario usar cuadros de textos de información, error, advertencia, entre otros de la librería Tkinter.

- Opciones de Labels, Buttons, Entrys, ComboBox, ventanas, entre otros.

.title (): Esta opción le permite ponerle un título a la ventana, debe ser en formato string y entre paréntesis. Por ejemplo: `ventana.title ("Hola")`.

.geometry (): Esta opción le permite definir el tamaño de la ventana ancho x altura en formato string. Por ejemplo: `ventana.geometry ("500x100")`.

.maxsize (): Esta opción le permite definir el tamaño máximo que se puede expandir la ventana. Por ejemplo: `ventana.maxsize (ancho, altura)`.

.iconbitmap (): Esta función le permite al programador agregarle un icono a la ventana, debe ser en formato string y el archivo en formato ico. Por ejemplo: `ventana.iconbitmap ("Hola.ico")`.

Relief =: Esta opción le permite agregarle relieve a los componentes de tkinter como botones, etiquetas, campos de texto, entre otros. El relieve se debe poner todo en mayúscula. Por ejemplo: `relief=FLAT`.

Width=: Esta opción le permite definir el tamaño del elemento en dimensión a X. Por ejemplo: `width = 20`.

Height=: Esta opción le permite definir el tamaño del elemento en dimensión a Y. Por ejemplo: `height = 50`.

Bg=: Esta opción le permite al programador definir el fondo de color de los componentes de Tkinter. Se debe poner el color en formato string y en minúscula. Por ejemplo: `bg = "blue"`

Fg=: Esta opción le permite al programador definir el color de letra ya sea en botones, campos de textos, etiquetas, etc. Por ejemplo: `fg = "blue"`.

Font=: Esta opción le permite al programador definir el tipo de letra, familia de letra. Entre strings el tipo de letra y separado por coma el tamaño. Por ejemplo: `Font = ("Helvetica", 10)`

Image=: Esta opción le permite al programador definir una imagen o agregar una imagen a un elemento, puede ser usados en botones, etiquetas, ventanas, etc. Se le asigna la variable que contiene la imagen a la función `image`. Por ejemplo: `Image = Salud`.

Command=: Esta opción le permite al programador definir la función que desea realizar el botón. Se le debe asignar la palabra `command` a una función. Por ejemplo: `command=saludar`.

Row=: Esta opción le permite ubicar elemento en la posición de la fila que desea empezando desde 0 oeste a este. Esta función solo se usa con `.grid()`. Por ejemplo: `grid (row=1, column=0)`.

Column=: Esta opción le permite ubicar elemento en la posición de la fila que desea empezando desde 0 norte a sur. Esta función solo se usa con `.grid()`. Por ejemplo: `grid (row=1, column=0)`.

Text=: Esta opción le permite mostrar texto en un botón, etiqueta, etc. Debe asignarle a la variable `text`, el string que desea que se muestre. Por ejemplo: `text = "Alo"`

Textvariable=: Es un instancia de `StringVar()` que se asocia con un texto en un botón, etiqueta, entre otros. Si la variable cambia, el nuevo valor será mostrado en el elemento. Por ejemplo: `textvariable= cambio`. Para obtener el valor de la variable se usa `.get()` y para poner un valor a la variable `.set()`.

3. Conclusiones

Se presentaron problemas con el cronometro, pero este se solucionó separando las opciones de label y la ubicación de este. Además otros problemas como calificar el juego que tenía que ser aleatorio y no dar pistas al jugador, este se solucionó creando una función en donde tenía que dar una lista con 4 valores diferentes que correspondía las posiciones de dicha jugada utilizando la función de python `random.randint()`.

Durante este proyecto se concluye que uno aprende por errores y entre más errores comete más aprende una persona, en este mundo no existe la perfección, sino que la perfección son los errores que cometemos y que vamos aprendiendo de este.

En este proyecto he adquirido conocimientos acerca de módulos de archivos y como desarrollar un juego a partir de interfaz gráficas. Además he adquirido un poco más de conocimiento acerca de la librería Tkinter.

4. Estadística de tiempo

Actividad Realizada	Horas
Análisis de requerimientos	2
Diseño de algoritmos	15
Investigación de ...	4:30
Programación	20
Documentación interna	3
Pruebas	6
Elaboración del manual de usuario	1
Elaboración de documentación del proyecto	1:30
Etc.	6
Total	59 Horas

5. Rubrica de evaluación

Concepto	Puntos	Puntos obtenidos	% Avance 100/ % / 0	Análisis de resultados
Acceso al programa desde el pseudo-teléfono	2		100	
Menú	1		100	
Validación de datos	5		100	
Crear combinación	5		100	
Hacer jugada	15		100	
Calificar jugada	15		100	
Uso del cronómetro por juego	5		100	
Uso del cronómetro por jugada	5		100	
Uso del reloj	5		100	
Terminar juego	5		100	
Configurar	10		100	
Top-10 resumen	5		100	
Top-10 detalle	10		100	
Ayuda (incluye manual de usuario)	5		100	
Acerca de	1		100	
Salir	1		100	
Documentación interna	5		100	
TOTAL	100			

6. Anexos

- ARCHIVO mastermind2016top10.txt

Estructura de datos: listas, se encuentra acomodado de esta forma formado por 3 sublistas que representa la dificultad y dentro de cada sublista está compuesta por el string dificultad y una lista de listas de jugadores. Por ejemplo:

```
[ [ "Fácil", [ [ "Pedro", 12, ... ], [ "Juan", 12, ... ] ] ], [ "Medio", [ [ "Pedro", 12, ... ], [ "Juan", 12, ... ] ] ], [ "Difícil", [ [ "Pedro", 12, ... ], [ "Juan", 12, ... ] ] ] ]
```

- ARCHIVO mastermind2016configuración.txt

Estructura de datos: listas

Este es la forma de las listas de configuración: Donde abrá variables globales cómo los siguientes. Abrá una configuración de iniciación pero este será cambiada por el usuario y almacenada aquí.

```
[ "Fácil", reloj, posicion, Panel, Cronometro, CuantoTiempo, CuantoJugada ]
```

- ARCHIVO JugadasDMasterMind.txt

Estructura de datos: listas

Este archivo de texto almacena todas las jugadas hechas por la computadora, no se repiten sólo hasta llegar a 100 jugadas se pueden repetir. Tiene la siguiente estructura: una lista de sublistas.

```
[ [ 1, 2, 3, 4 ], [ 2, 1, 3, 4 ] ]
```