



Instituto Tecnológico de Costa Rica.

Campus Tecnológico Central de Cartago.

Escuela de Ingeniería en Computación.

Taller de Programación.

Prof. Ing. William Mata Rodríguez.

Programa 2

KAKURO.

Ledvin Manuel Leiva Mata – 2023071280.

Fecha de entrega: 28 de abril del 2023, semana 15.

1 semestre, 2023.

Contenido

Enunciado del proyecto	3
Temas investigados	3
Modulo Tkinter	3
Uso de command=lamda	4
Modulo time	4
Modulo pickle.....	5
Modulo random.....	5
Modulo os	6
Tipo de dato .dat.....	6
Variables de uso global.....	6
Programación dirigida por eventos.	7
Uso de interfaz gráfica de usuario (GUI: Graphical User Interface) en el desarrollo del software.....	8
Desarrollo de GUI con tkinter: biblioteca incluida en Python. tkinter se le considera el estándar de facto para la programación de GUI con Python. En esta parte hay que describir los elementos específicos de tkinter usados para este proyecto.....	9
Conclusiones del trabajo	10
Estadística de tiempos:	10
LISTA DE REVISION DEL PROYECTO	11
Referencias	12

Enunciado del proyecto

DEFINICIÓN DEL PROYECTO

Kakuro es un pasatiempo de lógica, tipo crucigrama matemático ya que en lugar de completarlo con palabras usando pistas verbales se completa con números usando pistas numéricas. Consiste en rellenar con números del 1 al 9 las casillas vacías de una cuadrícula. La cuadrícula la podemos ver como una matriz de dos dimensiones. Aunque la cuadrícula puede tener diferentes tamaños, para efectos de este programa vamos a trabajar una cuadrícula de 9 x 9, es decir 9 filas y 9 columnas. Cada fila y cada columna pueden estar divididas en grupos de casillas seguidas que se deben completar. En la parte izquierda de las filas y en la parte de arriba de las columnas aparecen números a los cuales se les llama claves numéricas. El objetivo es que la suma de los números puestos en las casillas de la fila sea igual a la clave numérica de la fila, y la suma de los números puestos en las casillas de la columna sea igual a la clave numérica de la columna. No se puede repetir un mismo número para las casillas de la clave numérica, ya sea a nivel de fila o de columna.

Temas investigados

Modulo Tkinter

El módulo Tkinter es una biblioteca estándar de Python que proporciona una interfaz gráfica de usuario (GUI) para crear aplicaciones de escritorio. Tkinter se basa en la biblioteca Tcl/Tk, que es un conjunto de herramientas para crear interfaces gráficas.

Tkinter permite crear ventanas, botones, cuadros de texto, etiquetas y otros elementos gráficos interactivos que se utilizan en las aplicaciones. Proporciona una serie de widgets y funciones para construir la interfaz gráfica y responder a las interacciones del usuario.

El funcionamiento de Tkinter se basa en el uso de una raíz de la ventana principal, que es el contenedor principal de todos los elementos de la interfaz. Se crean y configuran los distintos widgets (botones, etiquetas, etc.) y se organizan en el diseño deseado utilizando gestores de geometría, como el gestor de cuadrícula (grid), el gestor de empaquetado (pack) o el gestor de colocación (place).

Para crear una aplicación con Tkinter, se debe importar el módulo y luego seguir una serie de pasos que incluyen la creación de la ventana principal, la configuración de los widgets y la asignación de funciones para manejar los eventos, como hacer clic en un botón.

Tkinter proporciona una amplia gama de funcionalidades, como el manejo de eventos, la gestión de la entrada del usuario, la creación de diálogos y la personalización de la apariencia de la interfaz. Además, se pueden combinar las capacidades de Tkinter con otras bibliotecas y herramientas de Python para crear aplicaciones más complejas.

En resumen, Tkinter es un módulo de Python que facilita la creación de interfaces gráficas de usuario. Proporciona widgets y funciones para construir la interfaz y responder a las interacciones del usuario, permitiendo así desarrollar aplicaciones de escritorio interactivas.

Uso de command=lambda

El uso de `command=lambda` en Python se refiere a la creación de funciones anónimas (también conocidas como funciones lambda) que se asignan como comandos a ejecutar en respuesta a eventos, como hacer clic en un botón en una interfaz gráfica o presionar una tecla.

La función lambda toma uno o más argumentos separados por comas y devuelve el resultado de la expresión evaluada. Estas funciones son llamadas "anónimas" porque no se definen con un nombre como las funciones regulares.

En el contexto de la interfaz gráfica, el uso de `command=lambda` permite asociar una función lambda como el comando a ejecutar cuando ocurre un evento específico, como hacer clic en un botón. El código dentro de la función lambda se ejecutará cuando se active el evento correspondiente.

Módulo time

El módulo `time` en Python es una biblioteca estándar que proporciona funciones relacionadas con la manipulación y medición del tiempo. Permite trabajar con operaciones de tiempo, como obtener la hora actual, medir el tiempo de ejecución de un programa y realizar pausas o retrasos en la ejecución de un programa.

El módulo `time` se utiliza para interactuar con el reloj del sistema y proporciona varias funciones útiles.

En resumen, el módulo `time` proporciona una forma conveniente de trabajar con el tiempo en Python, permitiendo realizar tareas relacionadas con la medición del tiempo, pausas en la ejecución y conversiones entre formatos de tiempo.

Modulo pickle

El módulo pickle en Python es una biblioteca estándar que se utiliza para serializar y deserializar objetos Python. La serialización es el proceso de convertir un objeto en una secuencia de bytes, que puede ser almacenada en un archivo o transmitida a través de una red, y la deserialización es el proceso inverso, es decir, convertir la secuencia de bytes de vuelta a un objeto Python.

El módulo pickle es útil cuando se desea guardar objetos Python en un archivo para su posterior uso o enviar objetos a través de una red. Permite preservar la estructura y los datos del objeto, incluyendo su estado interno, lo que facilita la reutilización del objeto en otro momento o en otro contexto.

Es importante tener en cuenta que el módulo pickle solo es compatible con objetos Python y no se recomienda utilizarlo para serializar objetos que contengan código ejecutable o que sean sensibles a la seguridad, ya que la deserialización de objetos puede representar un riesgo de seguridad si se utiliza con datos no confiables.

Modulo random

El módulo random en Python es una biblioteca estándar que se utiliza para generar números pseudoaleatorios. Proporciona diversas funciones y métodos que permiten generar valores aleatorios, seleccionar elementos al azar y mezclar secuencias.

El módulo random utiliza algoritmos y generadores de números pseudoaleatorios para generar los valores. Estos algoritmos generan secuencias de números que parecen ser aleatorios, pero en realidad son deterministas y se basan en una semilla inicial.

Para garantizar que los valores generados sean realmente aleatorios, es posible establecer una semilla utilizando la función `seed(valor)`. Si se utiliza la misma semilla, se generará la misma secuencia de números pseudoaleatorios, lo que puede ser útil para reproducir resultados.

Es importante tener en cuenta que los números generados por el módulo random no son verdaderamente aleatorios, sino pseudoaleatorios. Esto significa que, aunque los valores parezcan ser aleatorios, en realidad son determinados por un algoritmo y una semilla inicial. Sin embargo, en la mayoría de los casos, los números generados por random son suficientemente aleatorios para la mayoría de las aplicaciones.

Modulo os

El módulo `os` en Python es una biblioteca estándar que proporciona funciones para interactuar con el sistema operativo subyacente. Permite realizar diversas operaciones relacionadas con la gestión de archivos, directorios, variables de entorno, procesos y otras funcionalidades del sistema.

Permite interactuar con el sistema operativo de manera portátil, lo que significa que las funciones funcionarán en diferentes sistemas operativos, como Windows, macOS y Linux, sin tener que preocuparse por las diferencias específicas de cada sistema.

El módulo `os` también proporciona acceso a otras funcionalidades avanzadas del sistema operativo, como la manipulación de variables de entorno, la ejecución de comandos del sistema y la gestión de procesos. Esto hace que sea una herramienta poderosa para la automatización de tareas, la manipulación de archivos y la gestión del sistema en general.

Tipo de dato .dat

El formato de archivo `.dat` no tiene una estructura estándar y su significado puede variar según el contexto en el que se utilice. Generalmente, el formato `.dat` se utiliza para representar datos sin procesar o binarios que no siguen un formato específico como CSV, XML o JSON.

El archivo `.dat` puede contener cualquier tipo de datos, desde texto sin formato hasta datos binarios codificados. Por lo general, estos archivos se utilizan para almacenar información que no está destinada a ser interpretada directamente por humanos, sino por programas o sistemas informáticos.

El funcionamiento del archivo `.dat` depende del programa o sistema que lo utilice. En muchos casos, los archivos `.dat` se utilizan como archivos de datos para aplicaciones específicas que necesitan almacenar información en un formato personalizado. El programa que utiliza el archivo `.dat` es responsable de interpretar y manejar los datos almacenados en él.

Variables de uso global

En programación, una variable de uso global es aquella que se declara fuera de cualquier función o bloque de código y puede ser accedida y modificada

desde cualquier parte del programa, ya sea dentro de funciones, clases u otros bloques de código.

Cuando se declara una variable como global, significa que su ámbito de visibilidad abarca todo el programa y no está limitada a un solo bloque o función. Esto permite que la variable sea compartida y utilizada por múltiples partes del programa sin necesidad de pasarla como argumento entre funciones o utilizar técnicas de alcance local.

Es importante tener en cuenta que el uso de variables globales puede tener implicaciones en la legibilidad y mantenibilidad del código, ya que cualquier parte del programa puede modificar su valor. Se recomienda utilizar variables globales con moderación y en situaciones donde su uso sea realmente necesario, optando en su lugar por el uso de variables locales dentro de funciones y el paso explícito de parámetros entre ellas.

Programación dirigida por eventos.

La programación dirigida por eventos es un paradigma de programación en el cual el flujo de ejecución de un programa está determinado por la interacción con eventos o sucesos que ocurren durante la ejecución del programa. En lugar de tener un flujo de ejecución lineal y secuencial, en la programación dirigida por eventos, el programa espera y responde a eventos específicos que ocurren en el entorno o en la interfaz de usuario.

En este enfoque, el programa define un conjunto de eventos a los que puede responder y define las acciones o funciones asociadas a cada evento. Cuando ocurre un evento, el programa ejecuta la acción correspondiente a ese evento. Los eventos pueden ser acciones del usuario, como hacer clic en un botón, mover el ratón o presionar una tecla, o pueden ser eventos generados por el sistema o el entorno de ejecución.

Uso de interfaz gráfica de usuario (GUI: Graphical User Interface) en el desarrollo del software.

Una interfaz gráfica de usuario (GUI, por sus siglas en inglés, Graphical User Interface) es una forma visual e interactiva de presentar información y permitir la interacción con un programa o software. Proporciona elementos gráficos, como botones, menús, ventanas y campos de texto, que permiten a los usuarios interactuar de manera intuitiva con el software sin tener que utilizar comandos de texto o recordar comandos específicos.

La GUI se basa en la representación visual de elementos y acciones que los usuarios pueden realizar. Estos elementos se presentan en forma de ventanas, cuadros de diálogo, botones y otros componentes visuales, y los usuarios pueden interactuar con ellos utilizando dispositivos de entrada como el ratón y el teclado.

Desarrollo de GUI con tkinter: biblioteca incluida en Python. tkinter se le considera el estándar de facto para la programación de GUI con Python. En esta parte hay que describir los elementos específicos de tkinter usados para este proyecto.

- **Ventana (Window/Frame):** La ventana es el contenedor principal de una aplicación GUI. Proporciona un área donde se colocan otros elementos de la interfaz, como botones, etiquetas o campos de entrada. La ventana se puede personalizar en términos de tamaño, título y otros atributos visuales.
- **Etiqueta (Label):** Una etiqueta es un elemento de texto que se utiliza para mostrar información estática en la interfaz gráfica. Puede contener texto, imágenes u otros elementos visuales. Las etiquetas son útiles para proporcionar instrucciones, descripciones o simplemente para mostrar información relevante.
- **Botón (Button):** Un botón es un elemento interactivo que se utiliza para realizar acciones cuando se hace clic en él. Se puede configurar un comando o una función que se ejecutará cuando el botón sea activado. Los botones son comunes para activar eventos como guardar, enviar o procesar información en una aplicación.
- **Campo de entrada (Entry):** Un campo de entrada permite al usuario ingresar texto o datos. Proporciona una caja de texto donde el usuario puede escribir información y luego se puede recuperar desde el código. Los campos de entrada son útiles para recopilar información del usuario, como nombres, contraseñas o comentarios.
- **Canvas:** Es un widget que proporciona un área de dibujo en la cual se pueden crear y manipular gráficos, como líneas, rectángulos, círculos, texto, imágenes, entre otros. Es una herramienta poderosa para crear visualizaciones personalizadas y elementos interactivos en una interfaz gráfica de usuario.
- **Método Grid:** Es una forma de organizar y posicionar los elementos en una interfaz gráfica utilizando el sistema de cuadrícula en Tkinter. Permite colocar los objetos en filas y columnas, creando una estructura de cuadrícula flexible. Al utilizar el método grid, se asigna a cada objeto una posición en términos de fila y columna dentro de la cuadrícula. Los elementos se pueden expandir o contraer según sea necesario para adaptarse al contenido o al tamaño de la ventana.

Conclusiones del trabajo

Durante el desarrollo de la interfaz gráfica de usuario utilizando Tkinter, surgieron diversos desafíos debido a la falta de experiencia previa en esta biblioteca. Sin embargo, a través de la investigación y la exploración de las diferentes funciones y características proporcionadas por Tkinter, fue posible comprender y utilizar eficazmente las herramientas necesarias para crear una interfaz funcional.

La familiarización con las funciones básicas de Tkinter, como la creación de ventanas, la disposición de widgets y la gestión de eventos, fue fundamental para lograr el desarrollo exitoso de la interfaz. Además, el uso de objetos como Button, Label, Entry y Canvas permitió añadir interactividad y elementos visuales atractivos a la interfaz.

Aunque se encontraron dificultades en el proceso, el aprendizaje y la experimentación con Tkinter proporcionaron una base sólida para el desarrollo de interfaces gráficas de usuario en futuros proyectos. La comprensión de cómo utilizar las funciones y herramientas disponibles en Tkinter permitirá abordar desafíos más complejos y crear interfaces más sofisticadas en el futuro.

En resumen, a pesar de los obstáculos encontrados durante el desarrollo de la interfaz gráfica utilizando Tkinter, se logró comprender y utilizar las funciones necesarias para crear una interfaz funcional. La experiencia adquirida en este proyecto sienta las bases para futuros desarrollos de interfaces gráficas de usuario, aprovechando las capacidades de Tkinter y mejorando las habilidades en el diseño de interfaces interactivas.

Estadística de tiempos:

Actividad realizada	Horas
Análisis del problema	1
Diseño de algoritmos	1
Investigación de...	2
Programacion	8
Documentacion interna	1
Pruebas	1
Elaboración del manual de usuario	1
Elaboración de documentación del proyecto	1
Etc.	
TOTAL	16

LISTA DE REVISION DEL PROYECTO

Concepto	Puntos	Avance 100%/0	Puntos obtenidos	Análisis de resultados
Opción Jugar: despliegue ventana de juego	20	100%	20	
Botón Iniciar Juego (incluye creación del archivo del Top 10)	15	100%	15	
Botón Deshacer Jugada	8	100%	8	
Botón Rehacer Jugada	8	100%	8	
Botón Borrar Casilla	2	100%	2	
Botón Borrar Juego	2	100%	2	
Botón Terminar Juego	5	100%	5	
Botón Top 10	5	100%	5	
Botón Guardar Juego	10	100%	10	
Botón Cargar Juego (incluye el despliegue del mismo)	10	100%	10	
Opción Configurar	5	100%	5	
Ayuda en el programa: Manual de usuario	5	100%	5	
Cronómetro o Timer tiempo real	5	100%	5	
TOTAL	100	100%	100	
Partes desarrolladas adicionalmente		100%		

Referencias

- Python Software Foundation. (2021). Tkinter - Python interface to Tcl/Tk. Recuperado de <https://docs.python.org/3/library/tkinter.html>
- Python Software Foundation. (2021). 7. More Control Flow Tools. Recuperado de <https://docs.python.org/3/tutorial/controlflow.html#lambda-expressions>
- Python Software Foundation. (2021). time — Time access and conversions. Recuperado de <https://docs.python.org/3/library/time.html>
- Python Software Foundation. (2021). pickle — Python object serialization. Recuperado de <https://docs.python.org/3/library/pickle.html>
- Python Software Foundation. (2021). random — Generate pseudo-random numbers. Recuperado de <https://docs.python.org/3/library/random.html>
- Python Software Foundation. (2021). os — Miscellaneous operating system interfaces. Recuperado de <https://docs.python.org/3/library/os.html>