Instituto Tecnológico de Costa Rica

Escuela de Computación

Carrera: Ingeniería en Computación

Curso: Taller de Programación

“Documentación del Programa 3: Futoshiki”

Profesor: William Mata Rodríguez

Estudiante: Jean Franco Quirós Jiménez

Fecha de Entrega: 29 de Junio del 2021.

Semestre: I 2021

Contenido

Enunciado del Proyecto

Enunciado

Temas Investigados

Importancia del software de gestión de versiones

El software de gestión de versiones es aquel que nos proporciona un método para conservar y manipular diferentes versiones de los programas y/o aplicaciones que se vallan a desarrollar. Este tipo de software es un elemento indispensable a la hora de darle mantenimiento a un programa pues nos permite comparar versiones de este en caso deuna falla inesperada o recuperarlo si este se dañó por alguna razón.

Usar software de control de versiones tiene una gran cantidad de ventajas como las expuestas por Shapiro (2020): crear respaldos, proporciona un historial de cambios, permite aislar una parte del proyecto para experimentar y posibilita que varias personas trabajen en este a la vez. Existen dos tipos de software de control de versiones:

- Centrado: Se basa en un servidor central al cual cada usuario envía todos los cambios que realice al trabajo.
- Distribuido: Se basa en un servidor que cuando recibe un cambio, envía una copia actualizada a todos los participantes.

Software de gestión de versiones utilizado

El software de control de versiones utilizado para este programa es Git, una herramienta utilizada para almacenar proyectos en repositorios que se maneja por ramas (Branches en inglés). Esto significa que para realizar el control, se crea una rama (una réplica independiente del código) sobre la cual se podrá trabajar con tranquilidad pues ocupar revisar algo se pueden comparar la rama con su origen. Una vez se está satisfecho con el código en la rama, es posible actualizar su origen, imponiendo los cambios de la rama sobre este o crear otra rama a partir de la esta si se desea conservar ambas versiones. Git es distribuido por lo que cuando

varias personas trabajan juntas, es importante proponer los cambios antes de implementarlos para evitar que la versión principal de programa cambie súbitamente y minimizar problemas de compatibilidad entre los cambios de diferentes usuarios

Características de tkinter usadas:

Botones

Los botones son de los widgets más versátiles tanto en su configuración como ensus aplicaciones, pues son las herramientas más prácticas para que el usuario realice una acción. A los botones se les puede asignar una acción sencilla utilizando el parámetro “command”. Ejemplo:

Botón = Button (widget maestro (donde se colocará este widget), (…,)command = print (“Hola”)).

Hasta se les puede pedir que lleven a cabo funciones enteras de cualquier complejidad agregando “lambda:” después del “command=” y luego llamando a lafunción deseada con sus parámetros.

Entry

Los widgets Entry reciben y/o despliegan un string de una sola línea que el usuariopuede modificar. Se crean al asignarle una variable y llamar a la clase “Entry (widget_maestro (,…)). Se pude obtener el string contenido en un Entry al llamar a su variable y agregar el método “.get()” al final. También se puede asignar un texto inicial para este usando el método “insert(posición_en_la_que_se_desea_insertar_el_string, “string_a_insertar”)”.

Frames

Los frames son widgets cuya función más común es albergar otros widgets y facilitar el acomodo de los elementos en una ventana o en otro frames. Se crean asignándoles un nombre de variable y utilizando la clase “Frame ()”.

Existe una variante los frames llamada Labelframes que funcionalmente son iguales pero crea un cuadro que representa el contorno del frame y puede recibirel parámetro “text=” para añadir un título.

Los frames son particularmente útiles para acomodar widgets utilizando los métodos de posicionamiento de .grid y .pack (que son relativos al espacio que seles brinde) en un espacio específico cuando la raíz va a contener varios otros elementos con posiciones fijas.

“Tk()”

“Tk()” es una clase de la librería tkinter que cuando se llama y se le asigna una variable crea una ventana cuyo espacio será donde se ubicarán todos widgets quereciban como su parámetro de ubicación el nombre de la raíz.

El método “nombre_de_la_raíz.mainloop()” mantiene el programa constantemente corriendo, evitando que que la ventana creada se cierre de manera inesperada. Es importante añadirlo al final después que se definieron todos los elementos y funciones que se requerirán e interactuarán con otros en la ventana.

RadioButtons

Los Radiobuttons son widgets útiles para cuando se necesita seleccionar una de varias opciones. Antes de crearlos es necesario establecer la variable que estos compartirán para luego crearlos (preferiblemente todos seguidos) para que sea más depurarlos y asegurarse el valor de cada uno sea único.

Widgets

Se le denominan widgets a la gran mayoría de objetos interactuables y a algunosque no lo son que se encuentran en la librería de tkinter. En la mayoría de los casos; para crear un widget, hay que asignárselo a una variable aunque no es estrictamente necesario para algunos como los Labels pero siempre se va a ocupar llamar el widget que se desea crear como una función y al menos pasarlecomo primer parámetro el frame o la ventana que los hospedará y ya sea despuésde cerrar los parámetros del widget o en una línea separada asignarle un métodode ajuste de geometría (“.pack()” los coloca respecto a los puntos cardinales o lo cecntra, “.grid()” lo coloca según una cuadricula y ocupa recibir un valor de fila y de columna y “,place()” permite colocar el widget en un punto exacto ubicándolo con coordenadas ”x” y ”y”) . Este programa utiliza principalmente el método grid()

Elementos Adicionales

subprocess

subprocess es una librería que nos permite abrir subprocesos y acceder a sus vías de input, output y errores y obtener su código (Python.org, 2021) y como demuestra (dcaraballo, 2021) podemos utilizar este módulo para desplegar PDFs en pantalla: “Si deseamos abrir un archivo PDF en el visor de PDF estándar…, puede usar el comando subprocess.Popen ([ruta], shell = True). Esto no abre un shell intermedio, sino que abre el PDF directamente en el visor.” Por lo tanto este es el método que utiliza el botón “Ayda” para desplegar el manual de usuario

Explicación de la Solución

Estructuras de Datos

El programa emplea las siguientes estructuras de datos:

- Partida: Tupla que contiene 3 tuplas que representan el tablero inicial y las restricciones horizontales y verticales, si las 3 tuplas están vacías al iniciar partida, carga una partida aleatoria
- Datos: Matriz que contiene los valores de todas las casillas del tablero, de aquí se obtienen los valores para realizar las validaciones de una jugada
- rest_hor y rest_ver: Lista de tuplas representando las restricciones horizontales y verticales respectivamente en un formato más manejable para realizar las validaciones
- Lista_jugadas y Lista_borradas: Listas de tuplas que contienen las jugadas realizadas hasta el momento. Cuando se realiza una jugada se agrega una tupla con el valor, fila y columna de la jugada a Lista_jugadas, cuando se borra una jugada mediante el botón se pasa la última jugada en Lista_jugadas al inicio de Lista_borradas y viceversa al rehacer una jugada. Cuando se realiza una jugada se borran los contenidos de Lista_borradas para evitar rehacer jugadas obsoletas y que podrían no ser válidas

Tablero de juego

El tablero de juego consiste en una matriz de botones que refleja los contenidos de

“Datos” que contiene los valores de todas las casillas. Cuando se realiza una jugada, el juego obtiene el valor del radiobutton seleccionado a la derecha y realiza las comparaciones para determinar si la jugada es válida, si hubo algún incumplimiento, abre una caja para mensajes, especificando por qué no es válida y no la realiza; de ser válida, actualiza el valor en Datos y el texto en el botón correspondiente

Archivos

- “Documentación_Futoshiki.pdf” corresponde a este mismo documento
- “Manual_de_Usuario_Futoshiki.pdf” corresponde al manual de usuario
- “Futoshiki.py” es el programa principal
- “Futoshiki2021partidas” contiene 9 versiones diferentes de la estructura “Partida” separadas en tres listas con tres versiones cada una. Cuando se termina la configuración carga una versión al azar en “Partida” del nivel de dificultad seleccionado
- “futoshiki2021juegoactual” se crea al guardar una partida y contiene el estado de la partida en proceso cuando se presionó el botón “Guardar Partida”. Se sobrescribe cuando se guarda otra partida

Conclusiones

Problemas encontrados y sus soluciones

Al inicio se me complicó mucho lo que respecta al software de control de versiones y uno de los principales es que a la hora de instalar Git me solicitaba que ingresara un editor de texto para usar como predeterminado y como no había utilizado ninguno de los que me presentó (solo he usado el de python) no dejaba completar la instalación. Entonces procedí a explorar los editores de texto que me indicaba el instalador y terminé en Visual Studio Code y tras probarlo por un tiempo decidí utilizarlo. Luego me di cuenta que el instalador tampoco estaba funcionando correctamente, así que lo borré y volví a instalar y me dejó terminar de instalarlo.

Ahora estaba listo para empezar con Git y utilicé el taller del tutor para empezar pero cuando trataba de publicar algo, tanto desde Visual Studio como desde la consola, me impedía hacerlo debido a un problema de certificados. Buscando soluciones o alternativas

me enteré de que Gitub tenía una aplicación de escritorio y procedí a descargarla; por dicha funcionó y pude publicar los repositorios a a Github. Al final me quedo una cadena un poco complicada pero funcional: Escribía el código y realizaba los commits en Visual y en Github Desktop comparaba los cambios y publicaba los cambios y nuevas ramas a Github, donde realizaba las uniones.

Estadística de Tiempo

Actividad Realizada	Horas
Análisis del problema	4 h y 45 m
Diseño de algoritmos	6 h y 30 m
Investigación del software de control de versiones	14 h y 15 m
Investigación respecto a	13 h y 15 m
Programación	34 h y 45 m
Documentación interna	4 h y 15 m
Pruebas	10 h y 30 m
Elaboración del manual de usuario	5 h y 30
Elaboración de documentación del proyecto	11 h
TOTAL	96 h y 15 m

Repositorio de Git

Jean-FraQ/Futoshiki: Proyecto #3 de Taller de programación (github.com)

Lista de Revisión del Proyecto

Concepto	Valor	Puntos Obtenidos	Avance 100/%/0	Análisis de resultados
Opción configurar	6			No hay un timer que cargar. Actualmente al presionar iniciar partida se carga una partida al azar
Despliegue y manipulación de la ventana del juego: cuadrícula con sus restricciones y dígitos fijos. Incluye el despliegue de partidas.	12		100%	
Despliegue y manipulación de la ventana del juego: otros elementos	6		100%	
Botóm Iniciar juego	10		100%	
Crear Top 10	12		0%	No se pudo crear debido a la falta del reloj y timer
Botón Borrar Jugada	5		100%	
Botón Borrar Juego	2		100%	
Botón Terminar Juego	2		100%	
Botón Top 10	10		10%	El botón solamente existe pero no hace nada

Botón Guardar Juego	5		100%	
Botón Cargar Juego (incluye el despliegue del mismo)	15		100%	
Ayuda	5		100%	
Reloj Tiempo Real	5		0%	Falta de tiempo debido a un mal manejo
Timer Tiempo Real	5		0%	Falta de tiempo debido a un mal manejo
Total	100			
Partes desarrolladas adicionalmente				
Rehacer Jugada				Revierte la última jugada borrada mientras no se haya realizado otra. Se puede utilizar siempre y cuando Lista_borradas no esté vacía
Selector “Borrar”				“Borra” una casilla específica al jugar un espacio vacío en esta, El único impedimento para esta acción es que el valor de la casilla sea fijo

Referencias y Bibliografía

Costa, H. (4 de Octubre de 2018). *Widget Button (Botón)*. Obtenido de https://docs.hektorprofe.net/python/interfaces-graficas-con-tkinter/widget-button-boton/

Costa, H. (4 de octubre de 2018). *Widget Frame (Marco)*. Obtenido de https://docs.hektorprofe.net/python/interfaces-graficas-con-tkinter/widget-frame-marco/

Costa, H. (4 de Octubre de 2018). *Widget Radiobutton (Radial)*. Obtenido de https://docs.hektorprofe.net/python/interfaces-graficas-con-tkinter/widget-radiobutton-radial/

dcaraballo. (25 de Febrero de 2021). *¿Cómo abrir un archivo PDF en Python?* Obtenido de https://pythondiario.com/2021/02/como-abrir-un-archivo-pdf-en-python.html

Git.org. (s.f.). *About git-scm.com*. Obtenido de https://git-scm.com/site

Kumar, B. (4 de Junio de 2021). *Python Tkinter Grid (grid() method in Python Tkinter)*. Obtenido de https://pythonguides.com/python-tkinter-grid/#:~:text=Python%20Tkinter%20Grid%20Function%20Grid%20in%20Python%20Tkinter,games%20and%20applications%20like%20Calculator%2C%20Tic-Tac-Toe%2C%20Mines%2C%20etc.

Python.org. (28 de Junio de 2021). *subprocess — Subprocess management*. Obtenido de https://docs.python.org/3/library/subprocess.html

Recursos Python. (14 de Agosto de 2018). *Cuadros de diálogo (messagebox) en Tcl/Tk (tkinter)*. Obtenido de https://recursospython.com/guias-y-manuales/cuadros-de-dialogo-messagebox-en-tkinter/

Shapiro, J. (2020). Obtenido de https://www.youtube.com/watch?time_continue=30&v=xQujH0ElTUg&feature=emb_logo