

Alumno: Josué Gerardo Gutiérrez Mora.

Carné: 2018300436.

Asignatura: Taller de programación.

Grupo: 2

Profesor: William Mata Rodríguez.

TEC | Tecnológico
de Costa Rica

Contenido

Este proyecto se encarga de representar mediante programación en Python el juego Kenken 2.0 con todas las funcionalidades y reglas que este contiene, además de otras opciones para que el usuario pueda manipular de acuerdo a su gusto el juego agregándole nuevas mejoras a la versión anterior.

Cómo se menciona en el párrafo anterior, una de las opciones es la posibilidad de ingresar a un menú de configuración para que el usuario tenga la oportunidad de elegir la dificultad del juego, si desea que este tenga tiempo (o no tenga) o cronometro y si quiere que al ganar el juego tenga algún sonido o no además se ha implementado jugar con diferentes cuadrículas, un menú con 4 tipos de música, la posibilidad de guardar el juego en tiempo real y cargarlo no más iniciando la partida. Por otro lado cabe destacar que se muestran todos los tipos de posibles combinaciones para cada conjunto de casillas en el kenken.

También contamos en dicho programa con la opción “ayuda”, la cual nos mostrará un link con el pdf que contiene al manual de usuario. Igualmente se cuenta con la opción “Acerca de” que se encarga de mostrar la información del programa.

Enunciado del proyecto

Este programa es una modificación al programa número 2 (pasatiempo aritmético KenKen), de esta forma estamos trabajando con la actividad de mantenimiento de programas. Para este caso específico el mantenimiento se refiere a mejoras ya que se están agregando nuevas funcionalidades.

REQUERIMIENTOS DEL PROGRAMA MEJORAS (funcionalidades nuevas)

- 1- Los siguientes botones se agregarán a la opción de jugar: Este botón se puede usar en cualquier momento que el juego haya iniciado. Guarda en el archivo "kenken_juegoactual.dat" el juego con el estado actual incluyendo la configuración (nivel, reloj, etc.), nombre del jugador y jugadas realizadas para usar las funciones de deshacer y rehacer jugadas. El objetivo es que el jugador pueda en cualquier momento guardar el juego y posteriormente continuarlo. Este archivo solo va a contener una partida, la última que guardaron. En caso de que exista alguna partida en el archivo: se borra y se guarda la del momento. Este botón se puede usar solamente cuando un juego no se haya iniciado. Trae del archivo "kenken_juegoactual.dat" el juego que se haya guardado con su configuración y lo pone en la pantalla como el juego actual. El juego continúa cuando el jugador usa el botón de INICIAR JUEGO. GUARDAR JUEGO CARGAR JUEGO. Este botón se puede usar solamente cuando un juego se haya iniciado. Deshace o revierte la acción de poner o borrar un número en la cuadrícula. ¿ Cuántas jugadas se pueden deshacer ? Todas en forma secuencial según se hicieron, es decir, se puede seleccionar este botón cuántas veces se requiera mientras haya jugadas. Implementar esta funcionalidad como una pila de jugadas realizadas: push cuando se realiza una jugada, pop cuando el jugador pide deshacer una jugada. Este botón se puede usar solamente cuando existan jugadas. Rehace o reconstruye la última jugada que se deshizo. ¿ Cuántas jugadas se pueden rehacer ? Todas las que se hayan deshecho. Implementar esta funcionalidad como una pila de jugadas revertidas: push cuando se deshace una jugada, pop cuando el jugador pide rehacer jugada.

- 2- Posibles jugadas para una jaula. Esta funcionalidad sirve para desplegar al usuario todas las posibles combinaciones de jugadas que existen para que una jaula pueda completar la operación aritmética. Por ejemplo: si tenemos la operación $6+$ compuesta por una jaula de tres casillas en una cuadrícula de tamaño 5×5 debemos desplegar:

1, 2, 3

1, 3, 2

2, 1, 3

2, 3, 1

3, 1, 2

3, 2, 1

Los cálculos de las combinaciones disponibles deben ser dinámicos, es decir, se calculan en el momento considerando la operación, las casillas de la jaula y el tamaño de la cuadrícula.

3- Definir tamaño de la cuadrícula. o 3×3 o 4×4 o 5×5 o 6×6 o 7×7 o 8×8 o 9×9 o Multitamaño Con esta funcionalidad el tamaño de la cuadrícula la podemos cambiar: en el proyecto 2 la cuadrícula era de 6×6 , con esta mejora podemos definir los tamaños de cuadrículas indicados: desde 3×3 hasta 9×9 . Adicionalmente la opción multitamaño va a permitir que el jugador pueda ir avanzando automáticamente por los tamaños de la cuadrícula: empieza a jugar en el nivel 3×3 . Cuando logre terminar exitosamente un juego en este tamaño, el programa automáticamente lo envía a jugar al siguiente tamaño, es decir en el 4×4 y así sucesivamente hasta el último tamaño 9×9 . Al final de cada tamaño se determinará si va al Top-10 correspondiente. Cuando termina el último tamaño se queda jugando en dicho tamaño. Agregar a la opción Configurar este nuevo parámetro: Definir tamaño de la cuadrícula. Si el juego es multitamaño y se juega con Timer, el tiempo solicitado corresponde al tiempo máximo (horas, minutos, segundos) que se va a dar para completar el paso por todos los tamaños: desde 3×3 hasta 9×9 . El tiempo sugerido puede ser de 2 horas. El nivel de dificultad se mantiene en cada tamaño de cuadrícula, por ejemplo: si seleccionamos nivel fácil y multitamaño, cada tamaño se juega en nivel fácil.

4- Impresión de información A la opción del Top 10 agregar una opción para que la información de las mismas pueda enviarse a una impresora. Por ejemplo crear un archivo tipo PDF el cual permite la función de imprimir.

Marco teórico.

1) Tkinter: Esta es la biblioteca que trae Python por defecto para la creación de interfaces gráficas, trae muchas funcionalidades especializadas para la correcta programación de una GUI. A continuación, se mencionan las funciones de tkinter que se han utilizado en este trabajo:

- a) PyGame mixer: Mediante la utilización de esto, tkinter es capaz de reproducir sonidos en diferentes formatos a lo largo del video, pudiendo ser pausado o incluso detenido y volverlo a comenzar.
- b) Button: Función encargada de darle una variable las propiedades de un botón, este se encarga de llamar a una función para que ejecute a la susodicha.
- c) Label: Utilizada únicamente para mostrar texto en pantalla, tiene la posibilidad de ser actualizado en tiempo real según las funciones que se llamen.
- d) MessageBox: Su función principal es la de presentar un mensaje en una caja de texto de tal modo que informa al usuario de algún evento (ya sea que cometió un error, hacerle un recordatorio...).
- e) Entry: Las entradas funcionan para ingresar algún tipo de dato (int, float, string) que en un principio lo va a leer cómo un string, pero mediante programación y dependiendo de lo ingresado es posible que se lea cómo una variable de otro tipo.
- f) Frame: El frame es una representación de un trozo de interfaz gráfica donde podemos meter otros widgets, básicamente el uso del frame es para dar un mayor ordenamiento a los widgets y poder facilitar la colocación de los elementos en pantalla.
- g) After: Se utiliza dentro de una función para decir que esta se tiene que llamar a si misma (o a alguna otra función) cada cierto tiempo (dicho tiempo se da en milisegundos).
- h) Place_forget(): Mediante este método es posible desaparecer widgets de la pantalla.

2) Variables globales: Las variables globales son aquellas que pueden ser accedidas desde cualquier parte del código siempre y cuando se especifique que existen, además de que antes de mencionarla en alguna parte del código donde se supone que no existe se debe definir cómo global para así hacer que la función o la clase donde dicha variable esté metida reconozca que esa variable ya existe en alguna parte del código pero que ahora podemos acceder a ella.

3) Clases: Las clases sirven para facilitar el ordenamiento de las funciones y a su vez para crear objetos a futuro que se pueden utilizar para brindarle utilidades a una variable, por ejemplo, en este trabajo se ve implementada la utilización de una clase que es la encargada de actualizar los labels que muestran el tiempo del cronometro, una clase contiene partes fundamentales que en este caso serían:

- a) Self: Se antecede a las variables creadas para así indicar que a lo largo de la clase las funciones pueden acceder a dichas variables según corresponda y según el usuario haga el llamado de las funciones.
- b) Init: Es el constructor de cada clase, es lo que se ejecuta automáticamente apenas se llama a una clase.

4) Bind: Las llamadas bind a las funciones que son las encargadas de hacer uso de una función sólo en caso de cumplir con su específico evento.

5) Random: El módulo random que brinda muchas opciones de selección aleatoria, pero la utilizado en este caso para la selección de un juego según su tipo de utilidad es la randint(a,b) la cual permite seleccionar números enteros al azar entre a y b (Estos dos tomados en cuenta).

6) Archivos .dat: Encargados de guardar datos ya se fue de binaria o de forma regular (Escrito en castellano) sin embargo esta última un poco más complicada de utilizar debido a que el lenguaje debe implementar métodos para entender que significa cada elemento.

La imagen normal del kenken sigue siendo la misma que la del kenken 1.0, la diferencia es que a la hora de hacer las cuadrículas se ejecutan por medios diferentes, es decir, ya no sólo son de 6x6, si no que el usuario elige el tamaño.

Conclusiones

Entre los problemas encontrados se menciona el guardado de la configuración, pero la implementación del uso de listas facilitó drásticamente el guardado de los datos.

Aprendizajes adquiridos:

- Guardar una configuración.
- Aplicar el 'undo' and 'redo' en los elementos de programación.
- Aplicación de funciones de descomposición para la resolución de una fórmula matemática.

Actividad realizada	Horas
Analisis de requerimientos	1
Diseño de algoritmos	10
Investigación de	3
Programación	7
Documentación interna	1
Pruebas	1
Elaboración manual de usuario	1
Elaboración de documentación del proyecto	1
Etc.	2
Total	27

Concepto	Puntos	Puntos obtenidos	Avance	Análisis de resultados
Guardar juego	5			
Cargar juego	12			
Deshacer jugada	8			
Rehacer jugada	8			
Tamaños de las cuadrículas (3 puntos por 7 niveles) con sus respectivas jaulas y operaciones	21			
Multitamaño: avance por cada tamaño de la cuadrícula (2 puntos por 7 niveles)	14			
Impresión Top 10	10			
Posibles jugadas para la jaula	15			
Opción Configurar	5			
Ayuda	5			
Total	100			
Funcionalidades adicionales				