Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2024

PROYECTO 1

201901403 – Isamir Alessandro Armas Cano

Resumen

El código implementa una interfaz gráfica para la optimización de cambios de azulejos en un tablero. Define clases como Azulejo, NodoAzulejo, Piso, NodoPiso y Tablero, que representan la información del juego y su organización. La clase Tablero incluye métodos para cargar datos desde un archivo XML, mostrar tableros en la interfaz gráfica y generar un informe visual en formato de grafo. La aplicación también presenta una interfaz de usuario con opciones para mostrar carátula, leer un archivo XML y visualizar tableros. La aplicación se ejecuta mediante la creación de una instancia de la clase Aplicación.

Palabras clave

1. Azulejos
2. Tablero
3. Tkinter
4. XML
5. Grafos

Abstract

The code implements a graphical interface for optimizing tile changes on a board. It defines classes such as Tile, TileNode, Floor, FloorNode, and Board, representing the game's information and organization. The Board class includes methods to load data from an XML file, display boards in the graphical interface, and generate a visual report in graph format. The application also features a user interface with options to display a cover, read an XML file, and visualize boards. The application runs by creating an instance of the Application class.

Keywords

1. *Tiles*
2. *Board*
3. *Tkinter*
4. *XML*
5. *Graphs*

Introducción

El programa representa una aplicación en Python que aborda el desafío de optimizar cambios de azulejos en un tablero. Utiliza la biblioteca tkinter para la interfaz gráfica y otras bibliotecas para tareas específicas como la manipulación de archivos XML y la generación de gráficos. La estructura del programa incluye clases que modelan elementos como azulejos, pisos y el tablero en sí. El objetivo principal es cargar datos desde un archivo XML, mostrar visualmente tableros y generar un informe gráfico. La aplicación también presenta una interfaz de usuario con opciones para mostrar información sobre el desarrollador y leer archivos XML.

Desarrollo del tema

La aplicación en cuestión se enfoca en optimizar los procesos de cambio de azulejos en un tablero, y logra este propósito mediante la implementación de la biblioteca tkinter para la creación de una interfaz gráfica. Profundicemos en una exploración detallada de sus componentes clave y funcionalidades, desglosando cada aspecto para comprender mejor su arquitectura y utilidad.

1. Representación de Azulejos:

Dentro del código, se destaca la definición de una clase llamada "Azulejo", la cual cumple la función de representar azulejos individuales. Cada azulejo se caracteriza por su fila, columna, color y un booleano que indica si está volteado o no. Además, se introduce la clase "NodoAzulejo" para facilitar la creación de una lista enlazada de azulejos dentro de un piso, añadiendo una capa adicional de organización y estructura.

2. Estructura de Pisos:

La aplicación modela un piso mediante la clase "Piso", encapsulando atributos como su nombre, dimensiones (R x C), el número de volteos (F) y el tamaño (S). Los pisos se diseñan para acomodar patrones, que se almacenan como una lista de tuplas que contienen un código y el propio patrón. Esta estructura permite una gestión eficiente y organizada de la información relacionada con los pisos.

3. Estructuras de Datos Enlazadas:

Con el objetivo de optimizar la organización de datos, se implementan estructuras de datos enlazadas. La clase "NodoPiso" forma una lista enlazada de pisos dentro del tablero, y cada piso, a su vez, contiene una lista enlazada de azulejos. Esta jerarquía de nodos enlazados mejora la eficiencia y la facilidad de manipulación de la información.

4. Gestión del Tablero:

La clase "Tablero" desempeña un papel central al administrar la adición de pisos, cargar datos

desde archivos XML e iterar a través de los nodos enlazados. Para mantener la organización, se emplea un mecanismo de clasificación que ordena los pisos alfabéticamente según sus nombres. Esta estructura proporciona una gestión eficiente y ordenada del contenido del tablero.

5. Interfaz Gráfica de Usuario (GUI):

El código incorpora una interfaz gráfica utilizando tkinter, brindando a los usuarios una experiencia interactiva. La GUI muestra tableros iniciales y finales, permitiendo a los usuarios seleccionar y visualizar diferentes pisos. Además, ofrece la capacidad de generar datos en consola y presenta un informe visual utilizando la biblioteca graphviz.

6. Manejo de Archivos:

La aplicación demuestra sólidas capacidades de manejo de archivos al permitir a los usuarios cargar configuraciones desde archivos XML. Además, realiza validaciones de la estructura XML y emite mensajes de error apropiados en caso de problemas de análisis o inconsistencias en los datos, garantizando una robustez en la entrada de información.

7. Generación de Informes Gráficos:

Una característica destacada es la generación de un informe visual en formato de grafo. El código utiliza la biblioteca graphviz para crear una representación gráfica de los tableros, mostrando cada piso y sus patrones asociados. Esto proporciona a los usuarios una comprensión visual clara de la disposición y estructura del tablero.

8. Interacción del Usuario:

La GUI ofrece a los usuarios un menú con opciones para mostrar una portada, leer un archivo XML y salir de la aplicación. Esta interfaz permite a los usuarios navegar sin problemas entre diferentes funcionalidades, mejorando la usabilidad general y proporcionando una interacción intuitiva.

9. Información del Proyecto:

La clase "Aplicación" sirve como punto de entrada, configurando la ventana principal, creando una instancia de la clase "Tablero" e inicializando el menú de la aplicación. Este enfoque modular y estructurado facilita el mantenimiento y la expansión del proyecto, brindando una base sólida para futuras mejoras y adiciones.

Conclusiones

esta aplicación en Python aborda hábilmente la optimización de cambios de azulejos en un tablero mediante una interfaz gráfica intuitiva y estructuras de datos sofisticadas. Su flexibilidad, sólido manejo de archivos y generación de informes visuales la convierten en una herramienta versátil para explorar diversos escenarios de cambio de azulejos.

Referencias bibliográficas

1. Cardona Cañaveral, V. (2023). Ambiente de apoyo al aprendizaje de estructuras de datos en Python.
2. Roldán Blay, C. (2024). Uso de botones en aplicaciones de Tkinter en Python.
3. Roldán Blay, C. (2022). Desarrollo de menús y submenús con tkinter en python.
4. Moreno Madrona, N. Estructuras de datos dinámicas. Listas enlazadas.
5. Calvo Suárez, J. P. (2019). Introducción al manejo de sistemas de control de versiones con Git y Github.