
PROYECTO 1

202103718 1 – Harry Aaron Gómez Sanic

Resumen

Software desarrollado para simular el avance de una enfermedad, siguiendo un patrón de análisis estandarizado el cual fue brindado en el documento de la práctica, esta simulación puede ser realizada a pequeña escala analizando un tejido pequeño o por otra parte a gran escala en un tejido con miles de células.

Para el desarrollo de este software, se utilizaron listas simples y dobles enlazadas, para tener un mejor manejo de datos, creando las funciones necesarias para la simulación lógica y graficación de los resultados de cada uno de los pacientes. Además, se utilizaron distintas librerías para realizar estos procesos de manera más sencilla e interactiva con el usuario, como graphviz y fpdf.

El proyecto no cuenta con interfaz gráfica por lo cual se deberá interactuar con el por medio de consola ejecutando el archivo main, la graficación de los resultados se entrega al usuario en un archivo pdf y además se generará un archivo con extensión xml con los datos del paciente y el resultado de la simulación.

Palabras clave

Listas Enlazadas

Listas Doblemente Enlazadas

Librerías

Estructura de datos

Simulación

Abstract

Software developed to simulate the progress of a sickness, following a standard analysis pattern given in the document of the practice, this simulation can be conducted on a small-scale by analyzing a small tissue or in other hand on a big scale with in a tissue with thousands of cells.

For the development of this software, linked lists and double linked lists were used, to have a better data management, creating the necessary functions for the logic simulation and the graphics of the results of each of the patients.

In addition, different libraries were used to perform these processes on an easier and more interactive way with the user, such as graphviz and fpdf.

The project does not have a graphical interface, that is why you will have to interact with it through the console by executing the main file, the graphing of the results will be give to the user in a pdf file, also a file with xml extension will be generated with the patient's data and the result of the simulation.

Keywords

Simple linked list

Double linked list

Libraries

Data Structures

Simulation

Introducción

El proyecto consta de realizar un software que simule el avance de una enfermedad a través de un tejido matricial, que contenga un patrón inicial de células infectadas introducidas por el usuario, por medio de un archivo de entrada xml que se leerá y extraerá los datos.

Para realizar lo anterior descrito es necesario utilizar listas enlazadas simples y listas enlazadas dobles, con las cuales se manejan los datos a través de objetos auto referenciados llamados nodos. Además, es necesario hacer uso de la librería graphviz para representar gráficamente el avance de la enfermedad respecto a la cantidad de periodos solicitados, así como el resultado del análisis.

Además, para el análisis es necesario realizar un algoritmo que valide las células que estén infectadas y las que no para poder infectar las células vecinas y así determinar si el paciente padece de una enfermedad grave, leve o va a morir, imprimiendo un documento pdf en el cual se podrá visualizar cada uno de los periodos para poder determinar que tipo de enfermedad posee.

Desarrollo del tema

Al iniciar el desarrollo del programa se crearon 3 clases nodo, las cuales son, paciente, célula, patrón y rejilla, cada una de estas clases contiene atributos que se auto referencian. Las clases paciente, patrón y rejilla son clases nodo correspondientes a una lista enlazada simple respetivamente, estas listas enlazadas simples son los archivos, "Lista_Enlazada", "Lista_Patrones", "Lista_Rejilla_Patrones", estas listas contienen las funciones necesarias par el funcionamiento del software,

estas funciones se indicaran a continuación por archivo: • Lista_enlazada o Append():añade datos a la lista. o buscarPaciente(): Cuando el usuario selecciona un paciente en el menú principal, esta función busca y retorna los datos del paciente. o Print():Imprime los pacientes. o returnPaciente():Itera los datos de los pacientes y los retorna. o • Lista_Patrones o Append() o returnInfectadas():Imprime por consola la posición de las células infectadas o infectarSanas():Cambia el estado de las células. o returnInfectada():devuelve la lista de las células infectadas. • Lista_Rejilla_Patrones o Append() o returnListas():recibe la lista de los patrones de cada rejilla y la devuelve por consola. o Caso_1(): recibe el patrón inicial de la simulación, y compara las posiciones de las células infectadas en cada periodo, si una célula coincide con otra suma 1 a una variable auxiliar si la suma de esta variable es igual al tamaño del patrón de la lista el patrón es igual. o Caso_2(): Es también una función de validación con la excepción que valida entre patrones de cada periodo. Por otra parte, en la clase célula encontramos una clase nodo de una lista enlazada doble, la cual corresponde a la lista con nombre "ListaDobleEnlazada", cuyas funciones se explicarán a continuación: • ListaDobleenlazada o Append() o mostrarCelulas(): es una función que imprime por consola todas las células con estado 0. o returnCelula(): Función que al ser llamada retorna la lista donde se encuentran los valores de posición y estado. o graficarLista():esta función recibe los datos de nombre y edad del paciente así como la dimensión de la matriz y los periodos de simulación, crea una gráfica dentro del editor de archivos 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 2021. de Python, documento al cual se le asigna una variable y formatea el nombre, edad y periodo como cabecera de la imagen, seguidamente crea un nodo principal con la forma de caja, según las especificaciones de graphviz y con un bucle while se inicia el relleno de la gráfica, seguidamente de esto se valida si el estado de la célula a rellenar es 1 si es cierto la casilla a rellenar lo hará de un color más fuerte para distinguir las células infectadas de

las que no lo son, además se valida dentro de graphviz el tamaño de la matriz con un bucle for que recorre el dato de dimensión recibido por la función y grafica la cantidad de casillas necesarias para recrear la simulación. Por último, con la función open, se abre un archivo con extensión dot para escribir el contenido de la variable dentro de el y con la librería os crear un archivo imagen con extensión png. o filasArriba(): En esta función se valida la regla 1 dada en el documento del proyecto, donde si una contagiada tiene como vecino a 2 o 3 células contagiadas continua contagiada, para las filas superiores. o filasAbajo(): función parecida a la anterior con la diferencia que se validan las filas inferiores. o validacionReglaUno(): Se utilizan las anteriores 2 funciones para validar las posiciones de las células infectadas y dependiendo del caso se cambian los estados de la célula. o validacionReglaDos(): Utilizando las funciones de validación de las células, en esta función se valida si una célula sana tiene como vecinos a 3 células contagiadas, si es el caso se cambia el estado de la célula a 1 que es el estado contagiado. o comportamientoCélulas(): En esta función se recibe como parámetro la lista de los patrones iniciales, con estos patrones y los anteriores métodos de infección y desinfección, se valida la lista recibida como parámetro y se devuelve una nueva lista con nuevos patrones, los cuales serán graficados en su respectiva función. Para terminar, el software contiene un archivo principal llamado “main”, el cual reúne todas las listas y clases que contienen datos para realizar la simulación. Analizando el archivo, tenemos una función muy importante, la función para leer el archivo xml de entrada el cual recibe como parámetro la dirección o path del archivo que se quiere leer con esta ruta y con la librería etree de xml se asignan todos los valores de las etiquetas y de los atributos a unas variables antes declaradas, esto por medio de ciclos for con los cuales también se accede a etiquetas internas de otras etiquetas. Finalmente, al tener todos los datos del xml se le asignan esas variables a la clase paciente, la cual los recibirá como parámetros, y la clase paciente a la lista simple creada con anterioridad. Seguido de esta función se encuentran las funciones de menú por consola, las cuales

nos dan la opción a cargar un archivo xml, Seleccionar un paciente mediante un xml cargado con anterioridad y salir del bucle de ejecución. Si se selecciona la opción de cargar xml, se abrirá un submenú para introducir el nombre del archivo, si por otro lado ya hay un xml cargado, y se quiere seleccionar un paciente, al seleccionar esa opción se abrirá otro submenú en el cual se imprimirán todos los pacientes que se encuentren en el archivo xml y su edad respectiva y se dará la opción a seleccionar un paciente por número y al hacerlo iniciara el proceso de simulación y tráfico de los datos que se obtengan.

Seguidamente de esta función se encuentran otras 2 las cuales son convertirListaCélula y appendInfected las cuales van de la mano ya que la primera recibe la lista 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 2021. con los patrones iniciales, los valida con las funciones ya explicadas de la lista doble enlazada y devuelve una lista nueva con nuevos patrones y la segunda función tiene la funcionalidad de añadir o cambiar el valor de una célula dependiendo del patrón. La siguiente función tiene por nombre “generarPDF” y la funcionalidad que tiene es añadir las imágenes png que se generan en la función de graficarLista a un solo pdf y eliminar los documentos que la anterior función haya creado. Esta función fue creada por una problemática dada durante el desarrollo del software pues la graficación era desordenada y tosca. Por último, se encuentra la función de escritura de archivos xml, la cual recibe el nombre y edad del paciente, así como los periodos y el tamaño de tu matriz, también recibe el objeto de la lista con los patrones iniciales y la lista con los n patrones siguientes y los valida mediante la función de la clase “Lista_Rejilla_Patron”, dependiendo de lo que la función retorne se escribira diferente resultado en la etiqueta de resultado.

Conclusiones

El manejo de estructuras de datos es necesario para poder completar muchos de los proyectos de software que se presentan en la vida cotidiana, ya que estos permiten el manejo de datos de manera abstracta, lo cual facilita la utilización de estos en una solución o arquitectura de software que necesitemos o para algo

más simple como la utilización de una librería que no sea accesible con listas o arreglos nativos

En este proyecto, facilitó el uso de la librería graphviz y además dio la opción de generar funciones de resolución de resultados dentro de la matriz abstracta que se generaba por la introducción de los datos en el archivo de extensión xml.

Referencias bibliográficas

Bunch,J,R.. & Rose, D.J.(Eds.).(2014), Sparse matrix computations. Academic Press.

Craig Larman (2001), introducción al análisis y diseño orientado a objetos, Prentice Hall

David Budgen, (2003), Software Design, Second Edition, Pearson Addison Wesley

Luis Joyanes Aguilar, (2008), Fundamentos de programación, algoritmos, estructura de datos y objetos, McGraw-Hill

Apéndices:

