Proyecto Final Champions League

Luis Rodrigo Laguna García

Universidad Iberoamericana

Introducción

El programa demuestra lo aprendido en la materia de Estructuras de Datos,en donde se hace uso de manejo de memoria, con apuntadores, listas, pilas, colas, arboles binarios ,para este proyecto se hizo uso de un árbol binario degenerado,llamado así por solo crecer en una sola dirección.

Alcance

Este programa realiza una simulación de el torneo de clubes UEFA Champions League,haciendo uso de su formato de juego ,en este caso se usa en cada fase la eliminación directa para los equipos.Se desea implementar la mayor cantidad de conocimiento adquirido durante el semestre.El programa se desarrollo en el lenguaje Go(Golang),lo cual dificulto la codificación ya que este lenguaje es fuertemente tipado y en algunas ocasiones hacia parecer frustrante.Fuera de ello para el manejo de archivos resulto ser muy útil y eficiente.También se implementaron funciones de manejo de memoria para seguir con los temas vistos en el semestre,pero en GO no es necesario gracias a su garbage collector que ayuda a liberar memoria que ya no es utilizada.

El programa se estructura con un árbol binario degenerado,siendo la raíz un nodo vació,ya que se irán eliminando los equipos,cada equipo cuenta con tres estructuras enlazadas,una de ellas representa la información como tal del equipo,la segunda representa los datos de cada jugador del equipo y la tercera nos servirá para las estadísticas del equipo.Se implementaron dos librerías llamadas fileManagenent y convert,las cuales nos ayudan para ciertos aspectos.

FileManagement se implementa para el manejo de archivos,nos ayuda a leer el archivo en donde se encuentran los datos de cada equipo y jugadores,lo primero que se realiza es leer el archivo y convertirlo a un array de bytes,posteriormente ese array se convierte en uno de strings,delimitando su tamaño con el símbolo de (*).

Convert es una librería simple creada para convertir cada posición del array ya sea en tipo strig o int según se requiera para el llenado de datos del equipo/jugador.

Desarrollo

El programa depende de dos archivos de texto plano llamados plantilla.txt y jugadores.txt,aqui es de donde se extraerán los datos para llenar las estructuras de los equipos.Se crea tres estructuras para almacenar dicha información en la cual la primera pertenece a los datos del equipo,la segunda para los datos de jugadores y la tercera es para generar estadísticas de los encuentros.Se crean dos librerías,una para el manejo de archivos,leer el archivo,convertir a arreglos de strings los datos del archivo,también se crea una función para que sea mas fácil la obtención de los datos a partir separar los datos de cada equipo a partir de un símbolo,esta partición la convertimos en lineas con una función implementada,la otra librería se hace uso para convertir los arreglos en un tipo de dato especifico ya sea int o string según se requiera para el llenado de la información.

Dentro del programa principal se incorporan funciones para crear el árbol binario,la primera función utilizada crea un nodo raíz vació para facilitar el recorrido de los equipos a medida que estos son eliminados,posteriormente se realiza una función para crear los nodos siguientes que albergaran a los equipos con toda su información,a estos equipos se les asigna un numero(identificador) para que por búsqueda sean mas fácil de encontrarlos,cabe aclarar que el árbol binario que se esta creando es de tipo degenerado el cual va creciendo en un solo sentido a partir del valor cero incorporado en la raíz vacía.Una vez creados los nodos se implementan las funciones de encuentros en cada fase de el torneo,estas funciones generan números aleatorios para hacer la interactividad de jugadas y goles,se realiza una estructura de encuentro,la cual se basa que en primer instancia se juegue el partido y si no resulta un ganador se irán a tiempos extra,aun así,si no resulta un ganador se forza a que haya uno en fase de penales.Recalco que todas estas funciones de enfrentamientos son las mismas,solo cambia en el numero de iteraciones en que se va a ejecutar.Los equipos perdedores se eliminan del árbol con una función que realiza el eliminado del nodo considerando tres aspectos,uno si el nodo a eliminar no tiene hijos,si el nodo a eliminar tiene un hijo ya sea izquierda o derecha o si el nodo si tiene dos hijos,de esta forma se cubre que los enlaces sean re-conectados una vez sea eliminado el nodo.Al final del programa se manda llamar a la función de eliminar el árbol binario para liberar la memoria utilizada durante el programa.Entrando en el manejo de archivos,para poder escribir sobre el,se hace uso de ciertas funciones incorporadas en GO para crear el archivo y si existe sobre-escribirlo,una vez que el archivo ya no se utilice se debe cerrar,como buena practica.