Practica 2.2

Metodología i tecnología de la programación

Josep Miquel Jané Riba

Índice

Breve explicación del enunciado de la practica

Esta practica se basa en un programa para gestionar subastas tanto manuales, como digitales vía un archivo binario. Por otro lado, debe de poder mostrar los objetos en subasta activa o cerrada, además de el estado de las pujas, cuando cierras una de las pujas tiene que mostrar cual ha sido la ganadora, mostrando tanto el nombre como la cantidad.

La diferencia principal con la anterior práctica es que en esta nos pedían usar memoria dinámica, para no tener limite de pujas ni objetos.

Estructuración de la practica

Para esta practica he estructurado mi código de manera que tengamos un main muy simple con las diferentes opciones del menú.

Empezando por “loadInfo”, encargada de cargar la información del archivo indicado, con los datos de los objetos en subasta y sus precios, esta función devolvía al main el numero de objetos que se encontraban en el archivo.

Para poder continuar necesitamos preguntarle al usuario que opción deseaba, por eso he creado la función “showMenu”, que pregunta y devuelve al main, la opción deseada por el usuario.

A continuación, ya tenemos el menú, en el cual la primera opción que nos encontramos es la de mostrar la información del estado actual de cada objeto “showObject”, tal y como indica su nombre, la única función que tiene es mostrar el nombre de los objetos en subasta, acompañado del numero de pujas que tiene en ese momento.

La segunda opción que nos encontramos es la encargada de introducir pujas manualmente en el sistema “submitBid”, cuando ejecutamos esta función, el usuario a de introducir su nombre, el id del objeto que quiere hacer una puja, y por ultimo la cantidad de la puja que quiere hacer. Esta información se almacena en el array correspondiente.

Nos encontramos con la tercera opción, “getAutomatic”, consiste en una petición al usuario, en ella pedimos el nombre de un archivo que contenga en formato binario pujas para ciertos objetos. Las almacenamos y actualizamos nuestro array.

La cuarta opción es “showStatus”, en ella nos mostrara el estado de las pujas que hay en un objeto en especifico que nosotros queramos, nos mostrara todas las

pujas que ya hay, además de indicarnos con un '*', cual es la mayor de ellas. La diferencia de esta función respecto a la de la anterior práctica, es que las pujas que se muestran están ordenadas de mas grandes a mas pequeñeas.

Por último, nos encontramos "closeAction", que nos permite finalizar la puja de un objeto, junto a ello nos da el ganador de esta misma, a su vez no permite que se hagan más pujas.

Estas funciones son las mismas que había hecho en la anterior práctica, con la diferencia de que ninguna funcionaba con memoria dinámica, es decir lo único que he hecho ha sido modificar sus variables para adaptar el código, sin embargo, he añadido una función extra, "expansionBids", una función que me ha sido útil para poder reservar y modificar la memoria dinámica asignada a mis variables, para poder ir gestionándola durante todo el código.

Principales dificultades y soluciones

Las dificultades que me encontré en esta práctica fue adaptar el código, ya que se me hace mas complicado modificar algo ya hecho, que no empezar un código de cero. Tarde mucho en encontrar todos los sitios donde tenia que cambiar de variables estáticas a variables dinámicas, además de como reorganizar el código de algunas funciones.

Conclusiones

Me ha supuesto todo un reto poner en práctica todos los ejercicios de memoria dinámica, hubieses sido más fácil hacer el código de nuevo, pensándolo para hacerse con memoria dinámica. Al final he podido ir abriéndome camino a base de prueba y error.

Estimación de tiempo dedicado

Para el desarrollo de esta práctica, he destinado unas 12 horas.