LEGADO DEL SISTEMA

El primer problema que se manifestó fue el mal entendimiento de las funciones gráficas, por más que buscaba información no comprendía al 100 su utilidad. Creo que este fue el primer problema más grave que tuve, ya que no podía avanzar con el proyecto sin tener un buen dominio de ellas. Con el tiempo fue practicando algunas de ellas y pues eso me demoro algo en el desarrollo del proyecto.

Una vez que creamos el contenedor con arreglos tridimensionales y el uso de apuntadores para el manejo de memoria dinámica, el reto era jugar con los índices del contenedor, para poder desaparecer algunos cubos con el campo existencia, por un momento supuse que lo ideal sería desaparecer cada uno de ellos asignando las posiciones correspondientes, pero esto era algo muy tedioso y tardado ya que para un solo crucigrama de 35 palabras estar inicializando cada letra del contenedor, pues por ese camino no sería lo correcto, la única solución más factible fue, la de leer de un archivo de texto el crucigrama y guardarlo en una arreglo bidimensional, y así estar checando cada elemento del arreglo si existe alguna letra y si es así poner en existencia esa posición en el contenedor .

En lo que corresponde al manejo de archivo, no hubo mucho problema, el uso de ellos fue correcto para todos los casos de lectura y escritura de datos. Algunos detalles se fueron presentando pero no por el mal uso de ellos si no por no colocar las instrucciones de lectura en la parte correspondiente, eso me llevo al siguiente problema:

Una vez en el juego, ya que se haya capturado toda la información del jugador, el nombre y crucigrama a resolver, dentro de la función juega() hay un ciclo que controla los movimientos del cubo, dentro de esta función se encontraba un instrucción de lectura para la lista del crucigrama, pero por cada movimiento del cubo se realizaba otra lectura y así el programa se demoraba y pues realizaba asignaciones extras de memoria, esto conllevo a que después de algunos movimientos el programa se bloqueara y todo dejara de funcionar tanto el tiempo como los movimientos del cubo. La solución fue hacer la lectura fuera del ciclo y guardar las listas en una matriz y posteriormente solo tener una función de impremeLista () dentro del ciclo sin necesidad de hacer lecturas de archivos extras.

Al momento de hacer la comparación de la letra introducida por el usuario con el crucigrama necesitaba determinar una expresión para así hacer la convalidación correspondiente y tomar una decisión, trate de probar todas las comparaciones posibles pero esto era ineficiente. Para poder encontrar la forma de solucionar este problema, necesitaba determinar una relación entre el contenedor y la matriz ya que las posiciones de ambos arreglos correspondían a posiciones diferentes. En la matriz el renglón 0 del contenedor, le corresponde a la posición Ren -1 de la matriz, realice una análisis de la posiciones de ellos para arreglos de dimensiones más pequeñas y después más grandes para buscar la variación de índices y pues concluyendo con este análisis encontré que el índice de las columnas para ambas no cambia y así obtuve la siguiente comparación

if(cubo[0][i][j].letra[0] == mat[(nR - 1) - i][j])

Otro estrago en el proyecto fue al asociar uno de los elementos del proyecto, **el tiempo**. No tanto vincularlo al juego, si no más calcular el tiempo de transición, ya que utilizamos las funcione de time.h y pues al hacer el llamado de time() esta función retorna el tiempo en segundos del sistema del reloj del SO, que prácticamente es el tiempo en que fue creado el SO UNIX y hasta la fecha se ha llevado esta contabilización, el punto está en que debido a que mi juego consisten en ir decrementando el tiempo, necesitaba buscar la manera de cómo tratar el valor que me entregaba time(), esto es controlar el tiempo a mi disposición.

La única forma de solucionar este paradigma y que después me di cuenta que era muy fácil, fue hacer uso de algunas variables de tipo time_h y enteras, ambas variables fueron inicializadas con la función time(), excepto una cuya variable le agregue una constante k de incremento indicando el tiempo convertido a segundos solicitado por el usar y así determinar entre los segundos y minutos. Por último, sacaba algunas diferencias de tiempo y así obtenía el tiempo en forma decreciendo, cuando el tiempo era cero, actualizaba la llama con la constante para reinicia el proceso, esto fue tanto para calcular los segundos y los minutos transcurridos durante el juego.

Conclusiones:

El desarrollo de este proyecto fue algo muy completo al campo de programación, no tanto por el juego si por el uso de herramientas de programación que fueron utilizadas (Estructuras de control, estructuras de datos etc.) y sobre todo la implementación de una buena lógica para resolver los subproblemas presentados durante el proceso. Todo esto es abarcado en introducción a la programación, todo lo visto en el semestre anterior es aplicable a este proyecto, ya que es de esencia tener una buena base de esta materia. Durante el desarrollo del proyecto me di cuenta que para cada subproblema presentado existían una infinidad de solucione, creo la más correcta es la que te optimiza el programa, en términos de recurso de memoria y tiempo de ejecución ya que no son 200 líneas de código sin más de 1000 y si hacemos cuentas de todo el manejadero de información y el coste de tiempo esto es algo muy impactante. Estos factores se deberán de tener muy presente al momento de desarrollar todo tipo de programas.

Mis habilidades de programación han mejorado algo, son algunos detalles que hay que eliminar, pero el objetivo de este curso era el adquirir nuevas habilidades para el desarrollo y análisis de software y pues ha valido la pena este proyecto ya que inconscientemente las he adquirido.