**PROYECTO PRIMER PARCIAL**

**LISTAS ENLAZADAS**

Espinosa Jonathan

**Juego “Snake”**

Ingeniera Graciela Guerrero

Universidad de las Fuerzas Armadas “ESPE”

Listas enlazadas

Estructura de datos

Quito – Ecuador

2016

**Tema:**

Implementación del juego “Snake” con interfaz gráfica.

**Objetivo General:**

Implementar un juego “Snake” mediante una interfaz gráfica en visual Studio C++ para la utilización de conocimientos en listas simples o doblemente enlazadas.

**Objetivos Específicos:**

* Desarrollar un bosquejo de como irán las pantallas del juego.
* Consultar los debidos elementos del modo grafico en Visual Studio para poder implementar las pantallas del proyecto
* Implementar las clases con listas simples para empezar con el desarrollo del juego.
* Crear los métodos que ayudaran a que la serpiente siga el proceso que deseamos.
* Desarrollar funciones que me permitan trabajar con archivos planos para el ingreso y salida de datos.
* Evaluar los resultados obtenidos
* Corregir errores para fluidez del juego.

**Problema:**

¿Cómo implementar listas simples y doblemente enlazadas en el desarrollo del juego “Snake” en la materia de estructura de datos de la carrera de ingeniería en sistemas?

**Marco Teórico**

* **Modo gráfico en Visual Studio C++**

Visual Studio cuenta con un modo grafico en el que podemos crear interfaces con varios elementos como batones, textbox, etc. Los cuales cumplen una función determinada

* Vista Diseño:

Muestra la interfaz o el diseño visual del Proyecto, en esta vista se puede ver o modificar objetos de la superficie de diseño

* Ruta Navegación

La ruta de navegación nos permite desplazarnos rápidamente entre el modo de edición de plantillas, el modo de edición de estilos y el ámbito de edición de objetos

* Zoom

Este nos sirve para acercar o alejar la interfaz o los objetos que se encuentren en ella.

* Controles de la superficie de diseño

Existen varios controles entre ellos:

* Mostrar cuadricula de ajuste
* Ajustar a las líneas de la cuadrícula
* Activar o desactivar ajuste a las guías de alineación.
* **Juego Snake**

Es un juego que existió hace mucho tiempo y es por eso que hoy en día es considerado un clásico a nivel mundial, el cual consiste en que la serpiente o víbora sigue su recorrido tratando de alcanzar su comida con forme va comiendo va creciendo y la manera de perder es que choque con las paredes de la interfaz.

* **Archivos Planos**

Trabajar con archivos planos durante el proyecto tiene sus beneficios ya que nosotros requerimos que se guarde el puntaje y una vez cerrado el programa se vuelvan a cargar los datos el por eso que los archivos planos nos permiten hacer eso:

* **Crear el archivo plano:**

Creamos un fichero mediante un objeto de la clase ofstream y posteriormente lo leeremos mediante un objeto de la clase ifstream.

* **Manipulación de ficheros**

Toda transferencia de datos entre el programa y el fichero en disco se realiza a través del buffer.

* **Listas Simples**

Una lista simple cuenta con uno o varios nodos cada uno de estos contienen un dato y un puntero el cual va apuntando al siguiente dato de cada nodo y el último de estos apunta a null esa es la principal característica de las listas simples adicional a eso con las listas podemos realizar varios procesos como insertar elementos, eliminar elementos visualizar los elementos existentes en las listas etc.

**FUNCIONAMIENTO DEL PROGRAMA (JUEGO)**

Se realizaron 3 pantallas para la realización del juego de tal manera que el juego sea agradable para el manejo del usuario.

PANTALLAS

1. **Inicio**

**Funcionamiento:**

La pantalla de inicio permite la conexión con el resto de pantallas, tendrás 3 opciones (Jugar, Puntajes y salir) donde cada uno a excepción de Salir mostrara otra pantalla realizando su respectiva función.

**REALIZACION:**

Para su realización, simplemente se ubicaron imágenes al fondo y los botones para que tenga un aspecto agradable, el programa inicia mostrando esta pantalla y es de donde se llama al resto de pantallas.

1. Juego

**Funcionamiento:**

Dentro de la pantalla del juego tenemos a nuestro Snake moviéndose usando las teclas (w, a, s, d) en donde su único propósito será comer repollos para ir aumentando su tamaño poco a poco, dependiendo al computador donde se maneje el programa, el Snake se ira haciendo más rápido con respecto a que tanto se alimente.

**Choques:**

El Snake morirá únicamente cuando toque las paredes de la pantalla en donde se moverá

En caso de que el Snake, se coma a sí mismo, no morirá, simplemente el número de cuerpo (que se encuentra marcada en la parte inferior derecha del juego) disminuirá, su velocidad no se reducirá.

**Realización:**

Para la realización del juego, se aplicó Listas Simples, donde se guarda cada parte del cuerpo que se vaya aumentando.

Para el cuerpo se usaron Label que contengan imágenes y se mueva de acuerdo a las posiciones que se marcaran y cambiaran constantemente con el uso de un timer.

Los métodos usados para la realización del juego fueron los siguientes:

* 1. void iniciarJuego ();

Dentro de este método se declarará la iniciación del juego, en donde se dará la velocidad del Snake y se marcará inicialmente un Snake con 2 en número de cuerpo.

* 1. void crearGusano (int posx, int posy);

Haciendo uso de un Label auxiliar se creará en este método una nueva parte de cuerpo que será añadido a la lista de todo el gusano, asignándole sus respectivas posiciones que son recibidas como parámetros.

* 1. void crearComida()

En este método se darán posiciones ramdomicas que cogerá el label de comida, para darle una ubicación dentro de la pantalla de juego y ubicándola finalmente.

* 1. void perderJuego();

Dentro de este método se eliminan todos los Label y se libera la lista respectivamente, es decir se eliminan todos los labels creados y ya usados en el juego, posteriormente se realiza la multiplicación para sacar el puntaje y se cierra el formulario de la pantalla de juego

**Los métodos realizados en la Lista Simple para la realización del juego, fueron los siguientes:**

1. void ListaSimple::eliminarDato(Dato entrada);

Dentro de este método se eliminará la parte del cuerpo (Label) que este en una lista y contenga como Dato o Posición el valor de entrada

1. Nodo^ ListaSimple::buscarNodo(Dato destino);

En este método encontraremos el Nodo que contenga la parte del cuerpo que marca destino, es decir la posición donde estará el label que querramos encontrar.

1. void ListaSimple::insertarInicio(Dato entrada, Label^ pic);

El método de insertar, es aquel que nos deja ingresar la posición (entrada) y el label(parte del cuerpo) para luego mostrarla en la pantalla, dentro no está ordenado.

1. Label^ ListaSimple::ver(Dato i);

Junto al método de BuscarNodo, y el método ver nosotros podemos encontrar por medio de la posición que querremos mostrar, la parte del cuerpo o el Nodo donde se encontrara esa parte.

1. Puntaje

Respectivamente el cuerpo del Snake vaya aumentando el puntaje será mayor, y si se reduce el puntaje también se reducirá, puesto que el puntaje es igual al número del cuerpo por 173 para darle un aspecto más realista de juego.

Para el desarrollo del puntaje se utilizó archivos planos con el objetivo de que cuando el usuario juegue quede constancia de su puntuación y así ir comparando con otros usuarios, trabajar con ficheros nos dio la posibilidad de realizar un Backup de los datos antes ingresados así cuando se ejecute el programa se cargaran los puntajes obtenidos con anterioridad.

LIBRERIAS:

* #include <stdlib.h>
* #include <time.h>

Estas 2 librerías son usadas para el manejo de número Ramdomicos que nos darán la posición al azar de la comida del gusano.

* #include <stdio.h>

Esta librería es usada para entrada y salida de datos como por ejemplo la función printf(), scanf().

* #include <string.h>

definición de macros, constantes, funciones y tipos y algunas operaciones de manipulación de memoria.

* #include <iostream>

es utilizado para operaciones de entrada/salida. Su nombre es un acrónimo de Input/Output Stream.

# System.IO

La apertura y lectura de archivos para acceso de lectura es una parte importante de la funcionalidad de entrada y salida (E/S)

**Conclusiones:**

* Una vez realizado el proyecto implementando listas simples hemos podido observar que resulta más fácil realizar primero un bosquejo de lo que se va a realizar para con eso empezar a desarrollar poco a poco la interfaz.
* Después de haber analizado como se manejarán los datos para el puntaje de cada jugador pudimos darnos cuenta que con archivos planos es mucho más fácil recuperar los datos para que estos se queden guardados y se adhieran al momento de la ejecución.
* Al momento de desarrollar las pantallas pudimos ver que es muy importante hacer una consulta previa de cómo funciona cada elemento y de cómo son los comandos para que los elementos estén acorde a nuestras necesidades.

**Recomendaciones:**

* Para realizar un proyecto con listas ya sean simples, dobles o circulares se necesita dominar bien el tema para no tener inconvenientes durante el proceso de desarrollo.
* Para realizar un juego como el Snake se requiere de saber las reglas del juego o de cómo es el comportamiento de la serpiente.

# Bibliografía

*http://www.canalvisualbasic.net/*. (s.f.). Recuperado el 12 de 12 de 2016, de http://www.canalvisualbasic.net/foro/visual-basic-net/enviar-variable-de-un-form-otro-14523/

*https://msdn.microsoft.com/* . (s.f.). Recuperado el 12 de 12 de 2016, de https://msdn.microsoft.com/es-es/library/aa288422(v=vs.71).aspx

*La lista enlazada simple*. (s.f.). Recuperado el 12 de 12 de 2016, de http://es.ccm.net/faq/2842-la-lista-enlazada-simple

microsoft. (s.f.). *https://msdn.microsoft.com* . Recuperado el 12 de 12 de 2016, de https://msdn.microsoft.com/es-es/library/hh967574.aspx

microsoft. (s.f.). *https://msdn.microsoft.com* . Recuperado el 12 de 12 de 2016, de https://msdn.microsoft.com/es-es/library/jj620919.aspx

microsoft. (s.f.). *https://msdn.microsoft.com/* . Recuperado el 12 de 12 de 2016, de https://msdn.microsoft.com/es-es/library/hh279674.aspx

microsoft. (s.f.). *https://msdn.microsoft.com/* . Recuperado el 12 de 12 de 2016, de https://msdn.microsoft.com/es-es/library/hh279674(v=vs.110).aspx

http://www.nebrija.es/~abustind/Informatica/MetodologiaI/Archivos.pdf

ANEXOS

IMÁGENES

Cabeza del Snake: Cuerpo del Snake:

 

Comida del Snake:



CAPTURAS.



