

Práctica de Laboratorio de Programación III

Alumno: Juan Calderón Domínguez
DNI: 25098717J

Descripción del Juego:

El juego realizado es el clásico juego de la serpiente, en el cual la serpiente debe comerse todas las manzanas para pasar al siguiente nivel.

Requisitos:

El juego se ha probado en varias máquinas, desde un 486DX2 66Mhz con 12 megas de RAM y Windows 98, hasta un AMD XP 1800+, 256 megas de RAM con Windows 98 y 2000. Funcionando igual en todas ellas.

Como se ha programado:

El juego se basa en la programación de Tipos Abstractos de Datos (con tipos opacos), los cuales hemos utilizado cinco. Snake (se basa en la serpiente y sus funciones), Pantalla (relacionado con la pantalla), Escribir (librería que se encarga de mostrar textos en formato gráfico), Juego (Donde se cargan los datos de la partida) y Hiscore (donde se cargan y manejan los Records. Se utiliza otro módulo “CONVER” pero no es un tipo abstracto, contiene una serie de funciones que se utilizan en varios módulos.

Descripción de los procedimientos:

MODULO SNAKE

DEFINITION MODULE snake ;

TYPE serpiente ;

PROCEDURE Crear(VAR a:serpiente);
Se encarga de Crear la serpiente

PROCEDURE Izquierda(VAR a:serpiente);
Mueve la serpiente una unidad a la izquierda

PROCEDURE Derecha(VAR a:serpiente);
Mueve la serpiente una unidad a la derecha

PROCEDURE Arriba(VAR a:serpiente);
Mueve la serpiente una unidad hacia arriba

PROCEDURE Abajo(VAR a:serpiente);
Mueve la serpiente una unidad hacia abajo

PROCEDURE Incrementar(VAR a:serpiente);
Incrementa la longitud de la serpiente en una unidad

PROCEDURE DibujarSerpiente(VAR a:serpiente);
Dibuja la serpiente por pantalla

```

PROCEDURE Muerde(VAR a:serpiente):BOOLEAN;
    Indica si la serpiente se ha mordido

PROCEDURE CoordinadasSerpiente(VAR a:serpiente;VAR x:CARDINAL;VAR y:CARDINAL);
    Devuelve las coordenadas x,y de la cabeza de la serpiente;

PROCEDURE UltimaCoords(VAR a:serpiente;VAR x,y:CARDINAL);
    Devuelve las coordenadas x,y de la cola de la serpiente;

PROCEDURE LongitudSerpiente(VAR a:serpiente):CARDINAL;
    Devuelve la longitud de la serpiente;

PROCEDURE Destruir(VAR a:serpiente);
    Destruye la serpiente

END snake.

DEFINITION MODULE pantalla;

TYPE PANTALLA;

PROCEDURE Refrescar(VAR a:PANTALLA;VAR nivel:CARDINAL);
    Refresca la pantalla en el monitor

PROCEDURE Crear(VAR a:PANTALLA;VAR nivel:CARDINAL);
    Crea la Pantalla desde un fichero según el nivel

PROCEDURE Destruir(VAR a:PANTALLA);
    Destruye la Pantalla

PROCEDURE PosicionPantallaValida(VAR a:PANTALLA; VAR x,y:CARDINAL):BOOLEAN;
    Devuelve Si la posición x,y es válida para la serpiente

PROCEDURE HayManzana(VAR a:PANTALLA;VAR x,y:CARDINAL):BOOLEAN;
    Devuelve si en la posición x,y hay una manzana

PROCEDURE PintarFondo(VAR a:PANTALLA;VAR x,y:CARDINAL);
    Sobrescribe el fondo en las coordenadas x,y.
END pantalla.


DEFINITION MODULE escribir;
TYPE LETRA;

PROCEDURE Escribir(a:CHAR;x,y:CARDINAL;letra:LETRA);
    Escribe una letra en pantalla en modo gráfico

PROCEDURE Cargar(VAR a:LETRA);
    Crea y Carga las letras desde fichero.

PROCEDURE Destruir(VAR a:LETRA);
    Destruye la variable donde se cargan las letras

PROCEDURE EscribirMucho(a:ARRAY OF CHAR;x,y:CARDINAL; VAR letra:LETRA);
    Escribe un Array de texto por pantalla en las coordenadas x,y.

PROCEDURE EscribirNumero(numero,x,y:CARDINAL;VAR letra:LETRA);

```

Escribe un numero de una cifra por pantalla en las coordenadas x,y.

PROCEDURE EscribirNumeros(numero,x,y:CARDINAL;VAR letra:LETRA);
Escribe un cardinal por Pantalla en las coordenadas x,y

PROCEDURE ScrollHorizontal(texto:ARRAY OF CHAR;x,x2,y,pause:CARDINAL;letra:LETRA);
Realiza un Movimiento de Scroll Horizontal de una cadena de texto en la altura definida como "y" y que empieza en x2 y termina en x

PROCEDURE ScrollVertical(texto:ARRAY OF CHAR;x,y,y2,pause:CARDINAL;letra:LETRA);
Realiza un Movimiento de Scroll Vertical de una cadena de texto en la posición horizontal "x" y que empieza en y2 y termina en y

END escribir.

DEFINITION MODULE Juego ;
TYPE JUEGO ;
RECORDS;

PROCEDURE Crear(VAR a:JUEGO);
Crea una partida

PROCEDURE PonerVidas(VAR a:JUEGO;vidas:CARDINAL);
Seleccionar el número de vidas

PROCEDURE QuitarVidas(VAR a:JUEGO);
Se le quita una vida al juego

PROCEDURE Vidas(VAR a:JUEGO):CARDINAL;
Devuelve el número de vidas

PROCEDURE SumarPuntos(VAR a:JUEGO;VAR puntos:CARDINAL);
Suma puntos en una partida

PROCEDURE Puntos(VAR a:JUEGO):CARDINAL;
Devuelve los puntos en una partida

PROCEDURE Nivel(VAR a:JUEGO):CARDINAL;
Devuelve el nivel del juego

PROCEDURE AumentarNivel(VAR a:JUEGO);
Aumenta el nivel del juego en un nivel

PROCEDURE Destruir(VAR a:JUEGO);
Destruye un juego.

DEFINITION MODULE Hiscore ;
TYPE ;
RECORDS;

PROCEDURE CrearRecord(VAR records:RECORDS);
Crea la variable Records

PROCEDURE CargarRecord(VAR records:RECORDS;fichero:ARRAY OF CHAR);
Carga los Records desde fichero

PROCEDURE MostrarNombreRecord(VAR records:RECORDS;posicion:CARDINAL;VAR nombre:ARRAY OF CHAR);

Devuelve el nombre del record que se encuentra en determinada posición

PROCEDURE MostrarPuntosRecord(VAR records:RECORDS;posicion:CARDINAL):CARDINAL;
Devuelve los puntos del record que se encuentra en determinada posición

PROCEDURE InsertarPuntosRecord(VAR
records:RECORDS;posicion:CARDINAL;puntos:CARDINAL);
Inserta los puntos del record que se encuentra en determinada

PROCEDURE GrabarRecord(VAR records:RECORDS;fichero:ARRAY OF CHAR);
Graba los records a un archivo

PROCEDURE DestruirRecord(VAR records:RECORDS);
Destruye los records

Los tipos son programados opacos y su descripción de cada uno es:

IMPLEMENTATION MODULE Escribir

TYPE

TLETRA=ARRAY[1..10],[1..10] OF CARDINAL;

LETRA=POINTER TO nodo ;

nodo=RECORD;

numeros: ARRAY['0'..'9'] OF TLETRA;

letras: ARRAY ['A'..'Z'] OF TLETRA;

puntos:TLETRA;

END;

IMPLEMENTATION MODULE Pantalla

TYPE PANTALLA=POINTER TO nodo;

nodo=RECORD

imagenArbusto,imagenTierra,imagenManzana:ARRAY[1..8],[1..8] OF CARDINAL;

tabla:ARRAY [1..pantallax],[1..pantallay] OF CARDINAL;

END; (* 1 seto 2 tierra 3 manzana *)

IMPLEMENTATION MODULE Snake

TYPE

serpiente=POINTER TO nodo;

nodo=RECORD

x,y:CARDINAL;

sig:serpiente;

END;

IMPLEMENTATION MODULE Hiscore

TYPE

nodo2= RECORD

nombre:ARRAY[1..8] OF CHAR;

puntos:CARDINAL;

END;

RECORDS= POINTER TO nodo1;

nodo1= RECORD

jugador1: nodo2;

jugador2: nodo2;

jugador3: nodo2;

END;

```
IMPLEMENTATION MODULE Juego
TYPE JUEGO=POINTER TO nodo;
    nodo=RECORD;
        puntos: CARDINAL;
        vidas: CARDINAL;
        nivel: CARDINAL;
    END;
```

Aspectos que resaltar:

Entorno grafico completo. Siempre se utiliza modo gráfico, en concreto se utilizan dos (640x480 16 colores) para el juego en sí y (320x200 256 colores) para el entorno gráfico. Las letras son cargadas desde fichero y se muestran en modo gráfico, componiendo frases desde caracteres a pantalla. Se han programado efectos con las letras como el Scroll vertical y el Horizontal. Los records son leídos y grabados en disco. Se realiza un chequeo de la velocidad del sistema para ajustar a la velocidad del equipo. Se aprovechan los tipos. Ejemplo: el modulo snake se utiliza en el juego y en la presentación. Cada vez que la serpiente se come una manzana, el marcador de puntos aumenta, dependiendo del nivel y de la longitud de la serpiente. El juego consiste en seis niveles cargados desde fichero.

Como se juega:

Para jugar se utilizan las teclas

A: Izquierda
S: Abajo
D: Derecha;
W: Arriba;
ENTER. Aceptar;

CONTENIDO DEL CD POR DIRECTORIOS

MANUAL: este documento en pdf
HD: el juego para jugarlo en el disco duro para poder guardar los records
HD/FUENTES: el código fuente del juego
CD: : el juego para jugarlo en el CD, no se guardan los records
CD/FUENTES: el código fuente del juego