

# Las maravillosas aventuras de Enrique la serpiente vegana y la manzana que se teletransportaba

11/04/2022

—Asier Aguilar, Ivan Calvo, Marcos Chouciño

## Visión general

**Semana 1:** Después de pensarlo mucho decidimos que con nuestros conocimientos sobre el funcionamiento de la NDS éramos capaces de hacer un juego al estilo del clásico “Snake”. Planteamos la idea general del proyecto además de las bases de su funcionamiento.

**Semana 2:** Nos dividimos el proyecto de la siguiente forma: Asier se encarga de la creación de los sprites y de los fondos además de su implementación en C; Marcos de el funcionamiento base de la serpiente y de la matriz en la que se mueve; Iván de implementarlo todo en C corrigiendo los múltiples fallos de sintaxis y de funcionamientos de las subrutinas.

**Semana 3:** Una vez hecho todo el trabajo asignado nos juntamos para arreglar los problemas de compilación surgidos por la unión de las partes del proyecto, ya que cada uno se encontró con distintos problemas y los arreglo de forma distinta.

**Semana 4:** El juego empieza a coger forma pero todavía tiene muchos errores, y a pesar de que se puede jugar, varias mecánicas básicas como que la manzana aparezca en posiciones lógicas o la longitud de la serpiente no funcionan bien.

**Semana 5:** La base del juego funciona bien, la serpiente crece y decrece como debe, la manzana aparece donde debe y los cambios de estado se dan cuando están previstos. Además se remodelan los fondos y se han cambiado ciertas cosas para que el juego funcione mejor.

## Objetivos

1. Adaptar el Snake para la Nintendo DS de una manera que sea compatible con aprobar la asignatura de Estructura de Computadores.

## Antes de leer el código

Durante el desarrollo del juego hemos hecho alguna reinterpretación del código para entendernos entre los integrantes del grupo por ejemplo hemos dividido mentalmente la matriz por “Bloques” siendo cada posición de la matriz uno de ellos y además hemos tratado de manera más comprensible para nosotros la presencia de los elementos sobre la matriz, diferenciando entre el alma y el cuerpo siendo el alma el estado de la posición de la matriz en la que se encuentra el cuerpo, es decir, los sprites.

***\*Esto puede resultar útil a la hora de leer el código.***

## Especificación del proyecto + descripción general del código

Hemos rediseñado el Snake de una forma que seamos capaces de programarlo con nuestros conocimientos actuales.

Antes de llamar al propio juego y establecer el primer estado se inicializan los gráficos y la paleta principal de colores, además de todas las interrupciones, y después se llama al juego.

Sin haber establecido todavía el primer estado se llama a las subrutinas que establece el vector de interrupciones, se habilitan las interrupciones del teclado, se configura el teclado habilitando las interrupciones de las cuatro teclas de la cruceta mediante el bit 14 del registro de control del teclado, se configura el temporizador calculando el latch para que interrumpa 2 veces por segundo, se pone en marcha el tiempo y se establece el primer fondo aparte de el estado a inicio.

### Inicio

El juego empieza en el estado INICIO en la cual tocando la pantalla táctil se cambiará el estado a JUEGO y el fondo a Fondo\_juego.

La detección de **la pantalla táctil** se realiza sobre toda la pantalla, no siendo necesario tocar una zona específica de la pantalla para cambiar de estado y comenzar a jugar.

Esta consulta de la pantalla táctil se realizará dentro del bucle while principal del juego, siendo esta una condición necesaria junto a que el juego se encuentre en el estado inicial para comenzar a jugar.

```
1 //Esta función consulta si se ha tocado la pantalla táctil.
2 int Pantalla_tocado() {
3     touchPosition pos_pantalla;
4     touchRead(&pos_pantalla);
5     return !(pos_pantalla.px==0 && pos_pantalla.py==0);
6 }
```

```
1 if(ESTADO == INICIO && Pantalla_tocado()) //al tocar la pantalla en cualquier ubicación.
2 {
3     //limpia la pantalla de arriba.
4     consoleClear();
5     //Manda generar la matriz e inicializar las variables globales.
6     configurarMatrizInicio();
7     //establece el fondo adecuado al estado del juego en ese momento.
8     visualizarFondo_juego();
9     //establece el estado a juego, (Estado en el que se desarrolla la partida.)
10    ESTADO = JUEGO;
11 }
```

Antes de comenzar con el estado Juego, se limpia el mensaje de la pantalla de arriba, se visualiza el fondo correspondiente a este nuevo estado y se configura la matriz principal por la que se va a desarrollar toda la partida, sin duda una de nuestras subrutinas más importantes ya que en esta se establecen las posiciones de los elementos así como el estado de cada posición y se inicializan muchas de las variables globales necesarias durante la ejecución del juego.

```
1 void configurarMatrizInicio() {
2     // Pone paredes, la serpiente y la manzana
3     HabilitarIntTempo();
4     int i;
5     int j;
6     int n;
7     for (i = 0; i <= 14; i++) {
8         for (j = 0; j <= 14; j++) {
9             escenarioMatriz[i][j] = TERRENO_NORMAL;
10        }
11    }
12    //esto coloca todas las posiciones con terreno normal
13
14    for (n = 0; n <= 14; n++)
15    {
16        escenarioMatriz[n][0] = PARED;
17        escenarioMatriz[n][14] = PARED;
18        escenarioMatriz[0][n] = PARED;
19        escenarioMatriz[14][n] = PARED;
20        //este bucle pone paredes al escenario
21    }
22    Cabeza_Fila = 6;
23    Cabeza_Columna = 10;
24    Cola_Fila = 6;
25    Cola_Columna = 10;
26    escenarioMatriz[6][10] = CABEZA;
27
28    l = 0;
29    Direccion = IZQUIERDA;
30    UltimaDireccion = IZQUIERDA;
31
32    memoriaReserva();
33    EstablecerPaletaPrincipal();
34    GuardarSpritesMemoria();
35
36    generar_manzana();
37    iprintf("\x1b[2;1H*INTENTA ATRAPAR LAS MANZANAS*");
38    // estos comandos ponen la serpiente en la pantalla, e inicializan las variables necesarias
39 }
```

## Juego

En este nuevo estado la serpiente empieza en el centrada a la derecha, mirando hacia la izquierda y se moverá hacia esa dirección una posición cada 0.5 segundos hasta que se pulse una nueva dirección mediante la cruceta de la consola. El funcionamiento de estas 4 teclas se tratará mediante interrupciones, en caso de pulsar una de las cuatro teclas el próximo movimiento de la serpiente será en la nueva dirección.

Como podemos observar en la **rutina de atención del teclado**.

```

1 //rutina de atencion del teclado que permite actualizar la direccion de la serpiente segun la direccion en la
  que se este desplazando en ese momento.
2 void RutAtencionTeclado()
3 {
4     if (TeclaPulsada() == ARRIBA && !(UltimaDireccion == ABAJO))
5     {
6         Direccion = TeclaPulsada();
7     }
8     else if (TeclaPulsada() == ABAJO && !(UltimaDireccion == ARRIBA))
9     {
10        Direccion = TeclaPulsada();
11    }
12    else if (TeclaPulsada() == DERECHA && !(UltimaDireccion == IZQUIERDA))
13    {
14        Direccion = TeclaPulsada();
15    }
16    else if (TeclaPulsada() == IZQUIERDA && !(UltimaDireccion == DERECHA))
17    {
18        Direccion = TeclaPulsada();
19    }
20 }


```

y la **rutina de atención del timer o temporizador**.

```

1 //rutina de atención del timer0 llama a las subrutinas necesarias segun la prosima posicion a la que se vaya a
  mover la serpiente.
2 void RutAtencion_timer0()
3 {
4     if(ESTADO == JUEGO)
5     {
6         if (EL_siguiente_es(Direccion) == TERRENO_NORMAL)
7         {
8             avanzar_cuerpo();
9             retroceder_cuerpo();
10        }
11        else if(EL_siguiente_es(Direccion) == MANZANA)
12        {
13            desaparecer_manzana(xaux,yaux);
14            avanzar_cuerpo();
15            l++; //longitud de la serpiente, equivalente a la putuacion.
16            generar_manzana();
17        }
18        else if(EL_siguiente_es(Direccion) >= 30000)
19        {
20            visualizarGame_over();
21            ESTADO = GAMEOVER;
22            consoleClear();
23            iprintf("\x1b[1;3H    ### GAME OVER ###");
24            iprintf("\x1b[4;3H*** PRESIONA <B> PARA ***");
25            tprintf("\x1b[6;6H** TU PUNTUACION **");
26        }
27        UltimaDireccion = Direccion;
28    }
29 }

```



El área jugable es una matriz de 15x15 posiciones, y 10 px por cada posición que se genera automáticamente al cambiar de estado INICIO al estado JUEGO. Tanto la primera como la última columna y fila serán pared por lo que quedan fuera del movimiento de la serpiente, dejando un área jugable de 130 x 130 píxeles. Y cada vez que la serpiente se coma una manzana, ésta desaparecerá, se creará otra en una posición aleatoria del mapa y la serpiente crecerá.

El movimiento de la serpiente depende de 5 subrutinas que establecen para cada posición de la matriz el estado que le corresponde y desplaza los sprites.

Dirección: El juego solo admite dos direcciones a cada momento, ya que tanto la dirección en la que se está mirando y la contraria no tienen ningún efecto (si estás mirando hacia arriba y pulsas abajo no pasara nada).

Movimiento: El movimiento de la serpiente se basa en borrar y generar sprites cada vez que el timer interrumpa. Primero borra la cabeza de la serpiente, pone un cuerpo en su lugar, genera una cabeza nueva (mirando en la dirección correspondiente) en la siguiente posición y por último borra el último cuerpo de la serpiente a menos que se coma una manzana. En este caso a parte de no decrecer por detrás, se generará una nueva manzana en una posición aleatoria libre (en la que no haya ni serpiente ni pared).

Game Over: Cuando la serpiente se choque contra una pared o su propio cuerpo se acabara el juego y se pasará al estado Game Over.

## Game Over

En este estado se cambia de fondo, se limpia la pantalla de arriba para mostrar un nuevo mensaje y se acaba el juego.

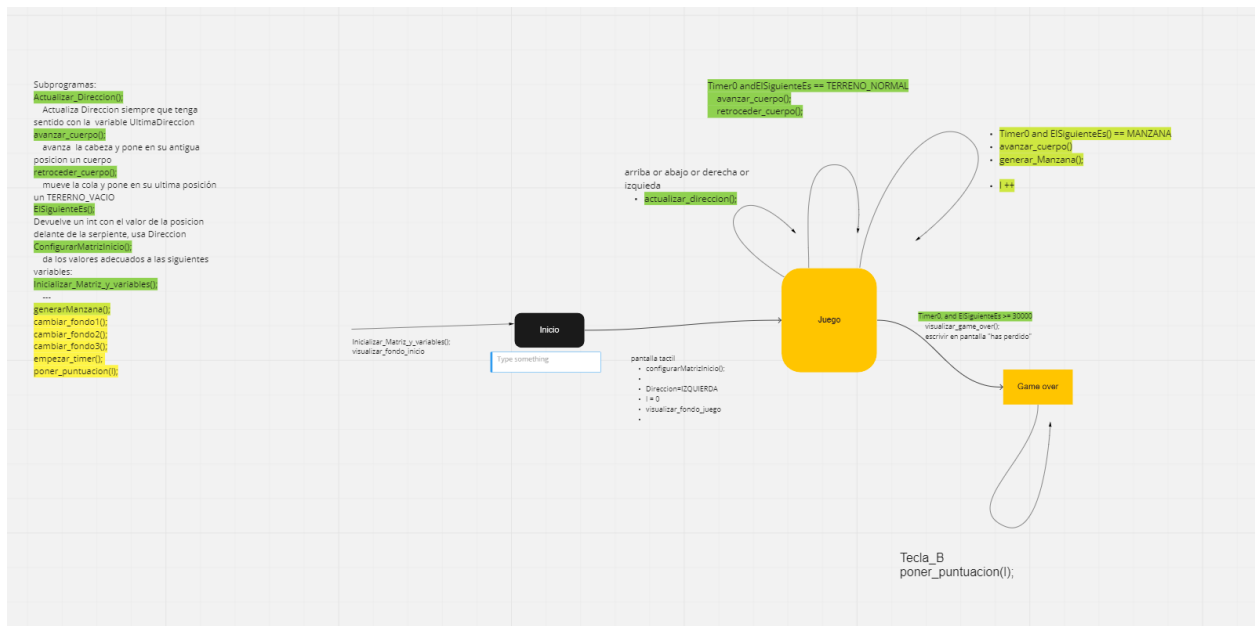
En este estado podrás pulsar la tecla <B> , **configurada por encuesta**, para comprobar tu puntuación (*la cantidad de manzanas que te has comido o la longitud de la serpiente sin contar la cabeza*) que se mostrará en la pantalla de arriba de la NDS.

## Esquema del autómata

### I. Cambios respecto el esquema original

Se ha modificado el esquema original por varios motivos. El primero es que no conseguimos borrar la serpiente para reiniciar el juego. La segunda es es que la tecla <B> se se consultará mediante encuesta para adecuarse a las especificaciones del proyecto

[https://miro.com/app/board/uXjVO8LgeDs=/?share\\_link\\_id=715265449130](https://miro.com/app/board/uXjVO8LgeDs=/?share_link_id=715265449130)



## Trabajo en grupo

A pesar de que desde el principio nos hayamos coordinado para hacer cada uno su parte y que luego funcione adecuadamente, nos dimos cuenta de que cada vez que juntábamos partes hechas por personas distintas, teníamos que arreglar ciertas contradicciones, ya que cada uno tenía su forma de afrontar los problemas. Aun así conseguimos que cada trabajo de los componentes del grupo funcionase de manera coherente, cohesionada y correcta. En conclusión, ha sido una muy buena experiencia ya que logramos dividirnos el trabajo equitativamente y todos logramos aprender tanto de nuestros errores como de las explicaciones que nos dábamos mutuamente.



## Conclusiones

Todos hemos adquirido nuevos conocimientos sobre el funcionamiento de C, la programación de videojuegos en la NDS y lo más importante, trabajar en equipo y aprender de las aportaciones que daba cada uno a la hora de resolver problemas.