

## II. Variables I

Has aprendido a crear un matamarcianos y ya deberías tener base suficiente como para explotar tu imaginación y añadir otros elementos como explosiones, distintos enemigos, incluso un segundo protagonista para que se trate de un juego para 2 jugadores, siempre que cuentes con los gráficos necesarios.

Pero todavía nos queda muchísimo por descubrir del lenguaje Benu. En este tema vamos a aprender a tratar conceptos un poco más abstractos de un videojuego, generalmente valores numéricos diversos como pueden ser la puntuación o los puntos de vida de un determinado proceso.

### II.1 Concepto de variable

Una variable es un pedazo de la memoria de nuestro PC en el que podemos almacenar un valor, generalmente numérico, aunque más adelante veremos otras posibilidades.

El ejemplo más claro son los aspectos básicos de un proceso en Benu como graph, size, etc. No son más que variables, simples números que se almacenan en la memoria de nuestro PC y a los que podemos hacer referencia con su nombre.

Todos los aspectos básicos de un proceso en Benu que hemos visto hasta ahora son útiles para crear un videojuego básico, pero no nos permiten recrear aspectos más complejos como la puntuación o los puntos de vida. Para ello necesitaremos crear nuevas variables, como veremos a continuación.

### II.2 Ámbito de una variable

Como ya hemos dicho, una variable puede almacenar datos como la puntuación o los puntos de vida. Es importante que observes que se trata de valores que se comportarán de manera muy distinta si tenemos en cuenta que la puntuación es un valor único, mientras que cada enemigo puede tener un valor de vida distinto...

Para salvar estas dificultades se utiliza el ámbito de las variables, que por ahora distinguiremos entre **global** o **privado**, según la siguiente definición:

**GLOBAL:** Una variable tiene este ámbito cuando almacena un valor único y común para todos los procesos del videojuego. El caso más claro es la puntuación del juego, ya que hay sólo una zona de memoria donde se almacena ese dato.

**PRIVATE:** Una variable tiene este ámbito cuando almacena un valor que puede ser distinto para distintos procesos. El caso más claro son los puntos de vida, ya que puede haber varios enemigos y cada uno tendrá una zona de memoria donde almacenar su valor de vida.

Este concepto nos permitirá inventar nuevas variables a las que podremos dar el nombre y los valores iniciales que queramos.

### 11.3 Variables GLOBAL

Las variables GLOBAL se incluyen en la zona de código que se sitúa al principio del programa, justo debajo de la sentencia PROGRAM y antes de la primera sentencia BEGIN.

Por ejemplo, para añadir una variable encargada de almacenar la puntuación, que tenga un valor inicial 0, nuestro programa quedaría así:

```
PROGRAM videojuego;

INCLUDE "DLL\import.prg";           //Incluye las DLL necesarias

GLOBAL
  puntos=0;
  ...

BEGIN
  set_mode(800,600,32,MODE_WINDOW); //Establecemos el modo de vídeo
  ...

END
```

Una vez incluida una variable GLOBAL, cualquier proceso es capaz de modificar su valor como si se tratase de cualquiera de sus aspectos básicos. Ejemplos de uso de la variable GLOBAL puntos serían los siguientes:

Puntos=100;	Asigna el valor 100 a la variable puntos.
puntos=puntos+1;	Suma 1 al valor de la variable puntos.
IF (puntos>1000) BREAK; END	Si la variable puntos almacena un valor mayor que 100 el proceso muere.

#### 11.4 Variables PRIVATE

Las variables PRIVATE se incluyen en la zona de código entre el PROCESS que encabeza la declaración de un proceso y el BEGIN que indica el comienzo de sus instrucciones.

Por ejemplo, para añadir una variable encargada de almacenar un valor de 100 en la vida inicial de cada enemigo, nuestro proceso enemigo quedaría así:

```
PROCESS enemigo();
PRIVATE
    vida=100;
BEGIN
    ...
END
```

Una vez incluida una variable PRIVATE, cada proceso es capaz de modificar su valor como si se tratase de cualquiera de sus aspectos básicos, siendo posible que distintos procesos conserven distintos valores. Ejemplos de uso de la variable PRIVATE puntos serían los siguientes:

vida=100;	Asigna el valor 100 a la variable vida del proceso.
vida=vida-1;	Suma 1 al valor de la variable vida del proceso.
IF (vida<=0) BREAK; END	Si la variable vida del proceso almacena un valor menor o igual que 0 el proceso muere.

### 11.5 Aplicando GLOBAL y PRIVATE

Crea la variable GLOBAL que almacena la puntuación y la variable PRIVATE que almacena la vida de cada proceso enemigo. Haz el uso adecuado de estas variables para conseguir que los enemigos sean capaces de resistir 10 golpes antes de morir, y que antes de morir sumen 1 a la puntuación.

Un ejemplo básico de enemigo que haga uso de las variables sería:

```
PROCESS enemigo();
PRIVATE
    vida=10;
BEGIN
    LOOP
        ...
        IF ( collision (type disparo) )
            vida = vida -1;
        END
        IF ( vida <= 0 )
            puntos = puntos + 1;
            BREAK;
        END
        ...
    FRAME;
END
END
```

No olvides crear la variable **puntos** en la sección GLOBAL, el hecho de crear una nueva variable se denomina formalmente **declarar**, y en adelante hablaremos de declarar nuevas variables, ya sean GLOBAL o PRIVATE cuando sea necesario.

Observarás que, a pesar de tener 10 puntos de vida, los enemigos mueren demasiado rápido... Esto se debe a que detectan la colisión durante varios FRAMES, y en cada uno de ellos restan vida puesto que es lo que hemos impuesto.

Más adelante veremos cómo resolver ese tipo de problemas.