

16. Tipos de Dato II

En el tema anterior aprendimos a crear variables de tipo vector, que no son otra cosa que listas de variables predefinidas como int, float o string, con la longitud que queramos.

En este tema vamos a continuar con los tipos de dato, aunque vamos a cambiar un poco el concepto, ya que ahora aprenderemos a crear nuestros propios tipos de dato en los que podremos almacenar información más compleja, como podrían ser las estadísticas completas del protagonista de un RPG, todo en una única variable.

16.1 Estructuras o types

Bennu nos ofrece la posibilidad de agrupar varias variables, sean del tipo que sean, incluso vectores, para crear un nuevo tipo de dato. Un tipo de dato con estas características se denomina estructura o type.

En un type podríamos incluir diversos aspectos, como por ejemplo una string con el nombre de un enemigo, un int con sus puntos de vida, etc.

La declaración de un type se realiza antes de la etiqueta PROGRAM, esto es, al principio del código, y su sintaxis es la siguiente:

```
TYPE <nombre_del_tipo>  
    <tipo1> <variable1> ;  
    <tipo2> <variable2> ;  
    ...  
END
```

Una vez declarado un type, el nombre del tipo puede utilizarse como prefijo a la hora de declarar variables, al igual que int, string y el resto de tipos de dato predefinidos. Es importante que veas la diferencia entre declarar un nuevo tipo de dato y declarar una variable asociada al nuevo tipo de dato, es un error común pensar que una vez declarado el type ya tenemos una variable, pero no es así, para tener una variable tenemos que declararla dentro de GLOBAL o PRIVATE, como hemos hecho hasta ahora.

Vamos a crear un nuevo tipo de dato donde almacenar unas estadísticas básicas para nuestro protagonista. Podemos dar el nombre que queramos al nuevo tipo de dato, aunque es recomendable que empiece por "tp_", ya que esto nos ayudará a distinguir el nombre del tipo de dato del nombre de las variables de ese tipo.

```
TYPE tp_estadisticas
    string nombre;
    int vida;
    int velocidad;
END
```

Observa que las variables que componen el type se declaran exactamente igual que el resto de variables que hemos definido hasta ahora en las secciones GLOBAL y PRIVATE. Las variables que componen un type se denominan formalmente campos, y pueden ser vectores o incluso otros types definidos anteriormente.

Por supuesto, también podríamos asignar un valor por defecto a los campos, por ejemplo:

```
TYPE tp_estadisticas
    string nombre = "Sonic";
    int vida = 100;
    int velocidad = 4;
END
```

Recuerda que hemos definido un nuevo tipo de dato, pero todavía no tenemos ninguna variable de ese tipo. Podemos añadirla en la sección PRIVATE de nuestro protagonista, quedaría así:

```
PROCESS protagonista ()
PRIVATE
    tp_estadisticas estadisticas;
    ...
BEGIN
    ...
END
```

¿Cómo accedemos a los campos de un type? Si recuerdas lo visto en temas anteriores sobre la variable father de todo proceso en Benu, utilizábamos el operador punto (.) para acceder a los distintos campos. La variable father es un type, y en todo type se accede así a sus campos.

```
<nombre_de_la_variable> . <nombre_del_campo>
```

16.2 Haciendo uso de nuestro nuevo tipo de dato

Modifica el protagonista de tu videojuego para que haga uso de los campos de la variable estadísticas y reduzca su campo vida al colisionar con los enemigos y se desplace a la velocidad que marca el campo velocidad. También puedes hacer uso de la función `write_var()` sobre `estadisticas.vida` para tener un marcador en pantalla donde mostrar sus puntos de vida restantes.

También puedes usar adecuadamente las funciones `write_var()` y `delete_text()` para conseguir que el nombre de Sonic permanezca siempre situado sobre nuestro protagonista, Por supuesto haciendo uso del campo `nombre` de nuestra variable `estadisticas`.

Para trabajar con más detenimiento todo lo aprendido sobre tipos de dato también puedes asociar a cada enemigo un `TYPE` con su nombre y sus puntos de vida, de forma que antes de comenzar a ejecutar su bloque de instrucciones dentro de `LOOP - END` asigne aleatoriamente su nombre y sus puntos de vida, y los muestre en pantalla.

Para ello puedes reaprovechar el `TYPE tp_estadisticas` o bien crear un tipo de dato nuevo.