

## Características generales de Juego Comando 1,9

Comando 1,9 (seguramente saldrán versiones posteriores), esta complemente realizado en Builder 6 C++.

Realizar la programación de un juego puede ser una tarea desafiante para el programador, pero también necesita tiempo y paciencia. Una misma acción en el programa del juego, seguramente puede involucrar distinta cantidad de instrucciones, pero es tarea del programador optimizar las mismas para mejorar el desempeño de su programa.

Comando 1,9, pretende ser un juego de acción y lograr entretener al usuario hasta cumplir los objetivos del mismo. Los objetivos del juego en esta versión es sumar una cantidad de puntos (30000) ganados en el combate contra objetivos armados. Los civiles no deben ser dañados, y hacerlo provoca un descuento importante de puntos acumulados.

El jugador, que asume el rol de un soldado, cuenta con un rifle con cargadores de hasta 50 tiros cada uno, además de granadas que debe cuidar en usarlas, porque tienen un amplio radio de acción, pudiendo dañar civiles, además de los objetivos armados.

Si el soldado no dispara a tiempo puede ser herido, tiene 4 vidas (los corazones). Pero puede recuperar vidas y/o municiones si dispara contra las botellas y cajas de armas que pueden aparecer ocasionalmente.

La estructura del juego se basa en 3 objetos imágenes (puertas) que en forma aleatoria van cambiando su contenido. La imagen que va apareciendo en las puertas depende de la variable numblanco, donde se almacena el resultado aleatorio. Valor que se sortea periódicamente controlado por el Timer6.

Como se usa en todo el programa, esta variable es GLOBAL.

Hay otras variables globales en el programa, se muestran a continuación:

```
//Defino aquí variables globales que usare en todo el programa  
  
int balas=50,cargadores=4,granadas=4,vidas=4,numblanco;  
int blanco1=0,blanco2=0,blanco3=0,dispara1=0,dispara2=0,dispara3=0;  
int enemigos=0,amigos=0,puntaje=0;
```

Nota: Como se aprecia todas tienen un valor inicial, es necesario en este programa.

Si la imagen es enemigo, civil, vidas o municiones es fundamental para la lógica, ya que de eso dependerá el puntaje a favor o en contra o si recupera vidas o si se reaprovisiona de armas el soldado. Dado que obtener vidas y armas depende del azar, solo una estrategia de economía de estos recursos puede ser favorable para el jugador, pero no garantizar su triunfo.

Para lo que sigue el programa utiliza dos eventos básicos que pueden ocurrir sobre los 3 objetos imágenes:

**ImagenMouseMove** (mover puntero sobre imagen)

**ImagenClick** (clic botón izquierdo sobre imagen)

**n es número de imagen**

Lo que siempre sucederá al hacer un clic, es que si hay balas entonces deberá aparecer una explosión sobre esa imagen y escuchar su sonido. Descontará la balas usadas (5 por disparo) y luego el análisis del estatus, es decir si elimino enemigos o amigos si puede continuar el juego (control de vidas y municiones), o alcanzo la meta final.

Es importante resaltar la diferencia entre un objeto **Image** y un objeto **UIImage**, si bien en ambos podemos colocar imágenes, en el objeto **UIImage** podemos cargar imágenes **PNG**, estas presentan la posibilidad de tener **fondo transparente**. Podemos convertir una imagen JPG en PNG con el mismo método y software con el que creamos ICONOS transparentes, solo se deben guardar como PNG en lugar de ICO.

Esta disponible el listado del programa junto con este documento, además del fuente en la carpeta de **Recursos** del **Tutor C**, para poder editarlo con el **Builder** y analizar en detalle el funcionamiento del mismo.

**RECUERDE:** *Realizar la programación de un juego puede ser una tarea desafiante para el programador, pero también necesita tiempo y paciencia.*