

Bucle principal y velocidad constante

Créditos

- » **Autor:** Hugo Ruscitti
- » **Fecha:** 9 de Marzo del 2005

Introducción

Habitualmente un videojuego realiza varias tareas invisibles al jugador: captura el estado de los controles, verifica colisiones, procesa eventos, altera estructuras, etc.

El bucle principal de un juego combina todas estas tareas y, generalmente, imprime en pantalla todos los cambios al mismo tiempo.

Esta idea permite que todos los objetos del juego parezcan moverse al mismo tiempo.

Diseño básico de un bucle principal

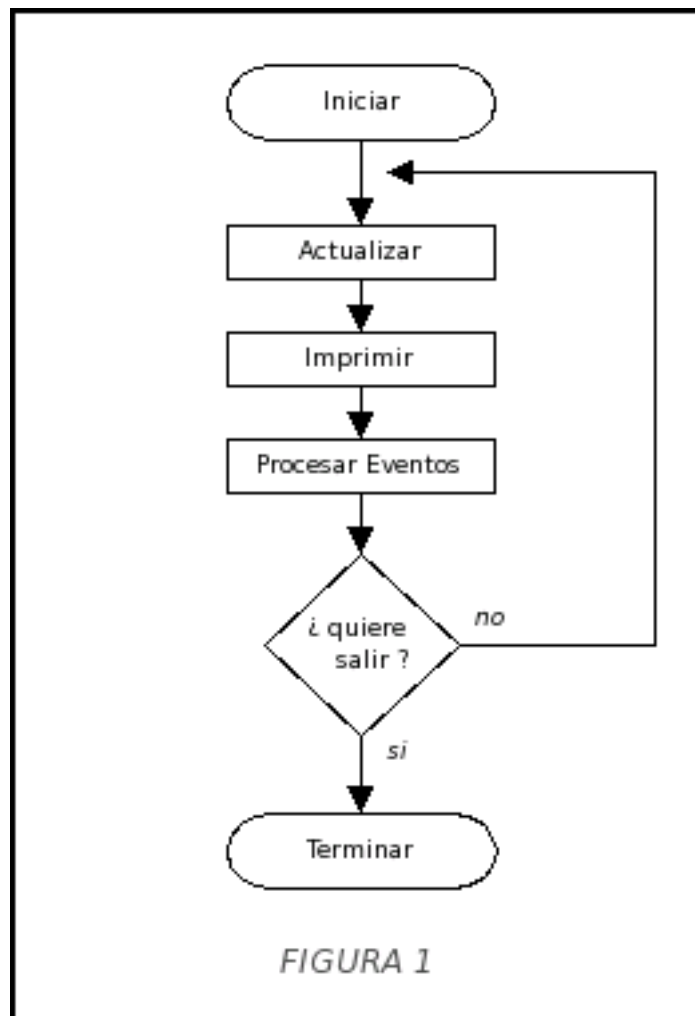


diagrama del bucle simple

La Figura 1 muestra un diseño de bucle principal muy simple, donde las tareas realizadas son:

- » actualizar: alterar la posición del objeto 'pelota'
- » imprimir: mostrar el objeto en pantalla
- » procesar eventos: verifica si el usuario quiere cerrar el programa

Para ilustrar este diseño muestro su implementación en lenguaje C utilizando las bibliotecas SDL.

- » Descargar ejemplos 1 y 2
- » Ver el código fuente del primer ejemplo



captura del ejemplo 1

El ejemplo 1 muestra una pequeña esfera rebotando por la pantalla.

Velocidad constante

Otra tarea importante que debe cumplir un bucle principal es garantizar una velocidad constante en diferentes tipos de equipos. El ejemplo anterior no cumplía este requisito, y debemos modificarlo.

Dado que cada equipo tiene diferentes velocidades, debemos basar nuestro diseño en un factor independiente a la arquitectura del equipo, el tiempo.

La biblioteca SDL tiene una función muy útil para el manejo de tiempo llamada `SDL_GetTicks`, que nos informa la cantidad de milisegundos transcurridos desde que iniciamos la biblioteca.

En el siguiente diseño debemos lograr que la función actualizar sea llamada la misma cantidad de veces por segundo en todos los equipos.

Para este ejemplo, la función actualizar, será llamada 100 veces por segundo. Mientras que la cantidad de llamadas a imprimir será variable en cada equipo, dependiendo de la velocidad del mismo.

En el mejor de los casos, la función imprimir podría ser llamada 100 veces por segundo (como máximo). La cantidad real de llamadas a imprimir nos permite medir el rendimiento del juego en cada equipo, y se lo suele

denominar FPS (frames per second - cuadros por segundo).

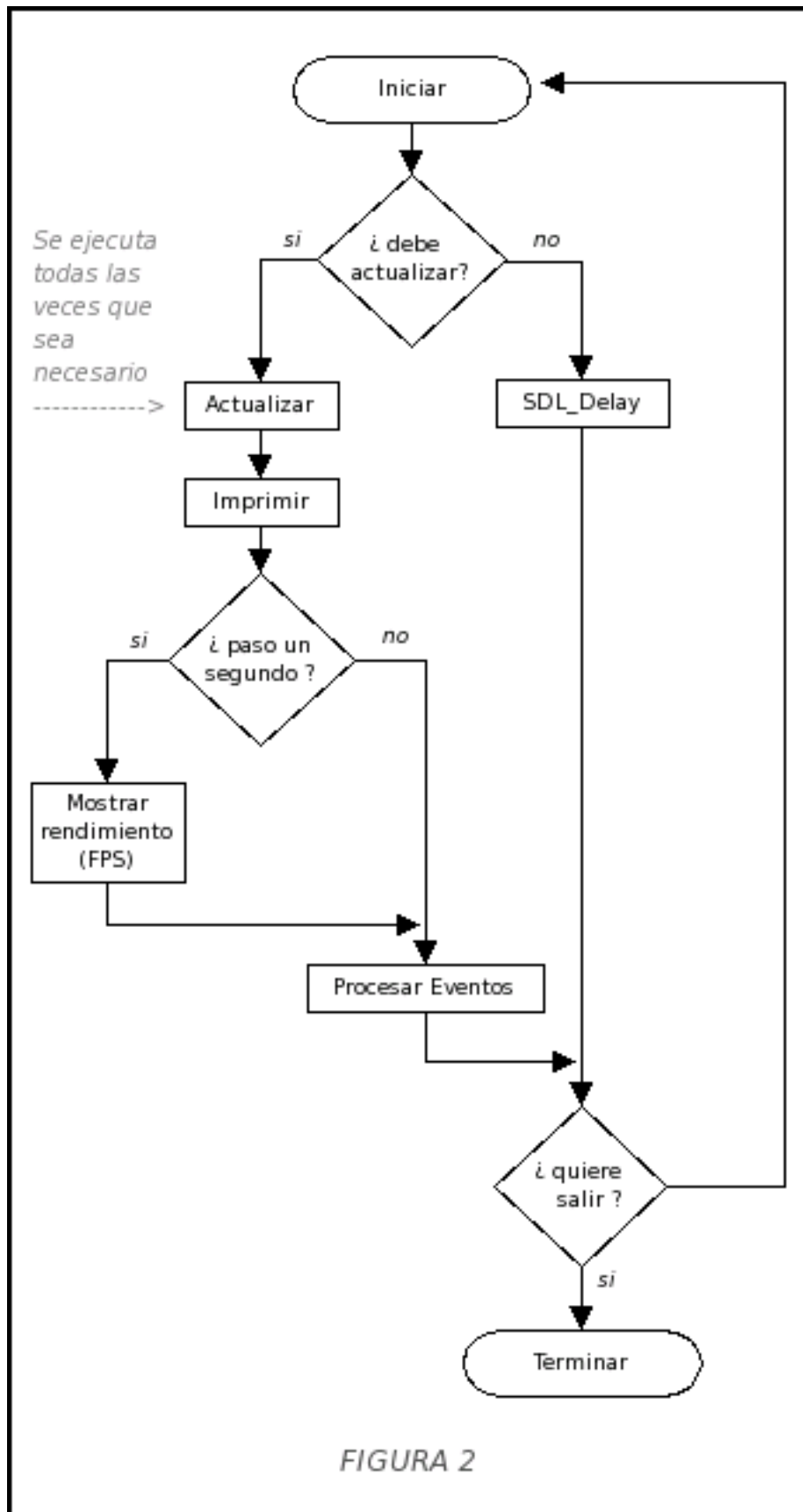


diagrama del bucle completo

La Figura 2 muestra una forma de aplicar este criterio, y su implementación se encuentra en el segundo ejemplo.

- » Descargar ejemplos 1 y 2
- » Ver el código fuente del segundo ejemplo



captura del ejemplo 2

Tenga en cuenta que este ejemplo, y el anterior, utilizan una técnica de impresión en pantalla muy sencilla y lenta, denominada Double Buffer. Si quiere mejorar este aspecto debe modificar la función imprimir utilizando otra técnica más eficiente como Dirty Rectangles, que se discute en otro artículo.

Conclusión

Construir un bucle principal es indispensable para comenzar un nuevo juego, y constituye el esqueleto principal del programa.

Es importante analizar su funcionamiento, cada bucle principal necesita estar adaptado a requisitos diferentes, y puede que este enfoque deba ser modificado por usted para su juego.

Muchas gracias.

Créditos

Se permite la copia, modificación y distribución de este artículo sólo bajo los términos de la GNU Free Documentation License

*Este documento ha sido generado automáticamente a partir del archivo 'bucle.xml' el Sat Dec 6 12:15:17 2008
La versión más reciente de este documento se almacena en www.losersjuegos.com.ar. Visitenos para obtener más recursos y actualizaciones.*