ALUMNO: Jorge Gangoso Klöck
GRUPO: Miércoles

Proyecto Final MP

Descripción

La práctica entregada consistía principalmente en ser capaz de modular correctamente un problema que trataba con memoria dinámica sin producir errores ni incurrir en pérdidas de memoria durante la ejecución.
Desde este punto de vista se trata de una práctica bastante sencilla ya que una vez creas un núcleo del programa que gestione correctamente la memoria dinámica los métodos adicionales son casi banales para el problema.
No obstante esto también quiere decir que si las bases del programa no están bien cimentadas puedes encontrar problemas de memoria mucho más adelante en métodos más complejos y no saber de donde vienen ya que el problema está en alguno de los métodos iniciales.
En mi caso fui muy precavido desde el comienzo por lo que me aseguré de perfeccionar los métodos básicos como el Redimensiona lo máximo posible para evitar esos futuros errores.
Creo que la parte obligatoria de la práctica es excelente aunque quizás sería algo más motivante si parte de la nota recayese, aunque en menor parte, de decisiones del alumno. Me refiero a las "tareas adicionales", si en lugar de dar unas pautas tan definidas de cómo el alumno debería ampliar el problema para alcanzar la máxima puntuación se dejara un poco más libre estoy seguro de que se harían propuestas más interesantes y motivaría más a los estudiantes a investigar, probar y encontrar nuevos errores y soluciones.

Ampliación

Como tarea adicional he escogido dos de las que venían propuestas en el documento, concretamente aquella que me pareció podría causarme más problemas y generar resultados más entretenidos de visualizar.

Primera Ampliación

En primer lugar hice que las particulas móviles al colisionar con fijas se clonasen (misma velocidad lugar aleatorio) si eran del mismo color o desaparecieran si eran de un color distinto.
Para lograrlo modifiqué el método Rebotes() dentro de la clase Simulador mediante una sentencia if-else(Fig. 1) muy sencilla una vez se ha comprobado que una partícula fija y una movil colisionan.

```
//HACER QUE DE LA VUELTA
 float ratiocambio=-1;
 this->moviles.ObtieneParticulaReal(i)->setdx(this->moviles.ObtieneParticula(i).getdx()*ratiocambio);
 this->moviles.ObtieneParticulaReal(i)->setdy(this->moviles.ObtieneParticula(i).getdy()*ratiocambio);
//APARECER Y DESAPARECER SEGUN COLOR
 //SI MISMO COLOR COPIAR
 if(this->moviles.ObtieneParticula(i).getcolor()==this->fijas.ObtieneParticula(j).getcolor())
 {
    Particula medionueva;
    medionueva.setcolor(this->moviles.ObtieneParticula(i).getcolor());
    medionueva.setdx(this->moviles.ObtieneParticula(i).getdx());
    medionueva.setdy(this->moviles.ObtieneParticula(i).getdy());
    this->moviles.AgregaParticula(medionueva);
 }
 //SI DISTINTO COLOR ELIMINAR
  else
 {
  this->moviles.BorraParticula(i);
 }
```

Fig. 1

Esto genera a veces una creación de particulas desorbitada si una partícula colisiona de forma estable y continua con un mismo objeto fijo del mismo color. Ésta es la razón de que escogiera añadir una segunda propuesta.

Segunda Ampliación

La segunda idea que tomé del documento orientativo es la de reiniciar el estado de la simulación pulsando una tecla, en mi caso “ESPACIO”.
Para esto simplemente debía modificar el bucle del main durante el que se ejecuta el programa comprobando si se pulsa la tecla “ESCAPE”, y añadir una comprobación para realizar la acción si se pulsaba ESPACIO.

En la imágen de la derecha (Fig. 2) podemos ver como al pulsarse dicha tecla se asigna parts (variable de partículas inicializadas para ser las móviles) y obstaculos (partículas con las velocidades a 0 para ser las fijas) a los respectivos atributos de la clase Simulador.

Cabe mencionar que tuve que crear una variable “t” para tecla() porque por alguna razón de buffer probablemente el programa dejaba de asimilar cuando pulsabas ESCAPE y únicamente se reiniciaba al pulsar ESPACIO. Una vez creada dicha variable el programa funciona correctamente realizando ambas acciones, tanto reiniciarse como cerrarse.

```
Simulador mySim (parts ,obstaculos ,ancho ,alto ) ;
int t= tecla();
    while (t != ESCAPE)
    {
    mySim.Step () ;
    mySim.Pintar (25) ;
        if(mySim.getFijas().getutiles()<20)
        {
            aux = mySim.getFijas() ;
            Particula p;
            aux.AgregaParticula(p);
            mySim.setFijas(aux);
        }
        if(t == ESPACIO)
        {
        mySim.setMoviles(parts);
        mySim.setFijas(obstaculos);
        }
    t=tecla();
    }
```

Fig. 2

Ésta parecio una opción muy interesante por varios motivos: primero podía deshacerme de esos cúmulos de partículas que se formaban a veces sin tener que matar el programa y aun así comprobar si había perdidas de memoria con valgrind tras varias ejecuciones seguidas. En segundo lugar me permitió comprobar que la aleatoriedad del programa funcionaba de manera bastante consistente ya que a pesar de repetir el estado inicial**O** del programa el resultado era diferente y esto era apreciable apenas segundos después del inicio de la ejecución.
Creo que tomé una buena decisión al hacer esta tarea ya que encontré un pequeño fallo en mi operador de asignación sobrecargado para la clase ConjuntoParticulas que probablemente habría pasado por alto de otra manera.

Main

Case 1:

Ésta es la salida de Valgrind para el caso 1 del main.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
Particula numero 1: Posicion X: 518.026  Posicion Y: 118.995Velocidad X: 6     Velocidad Y: 5Color: 4
Particula numero 2: Posicion X: 511.372  Posicion Y: 209.647Velocidad X: 9     Velocidad Y: 2Color: 4
Particula numero 3: Posicion X: 545.336  Posicion Y: 31.3235Velocidad X: 6     Velocidad Y: 4Color: 7
Particula numero 4: Posicion X: 562.253  Posicion Y: 43.8737Velocidad X: 9     Velocidad Y: 4Color: 2
Particula numero 5: Posicion X: 582.443  Posicion Y: 314.239Velocidad X: 0     Velocidad Y: 5Color: 7

El conjunto tiene: 10 Particulas.
Hay espacio para un total de: 10 Particulas
Particula numero 0: Posicion X: 512.026  Posicion Y: 113.995Velocidad X: 6     Velocidad Y: 5Color: 4
Particula numero 1: Posicion X: 518.026  Posicion Y: 118.995Velocidad X: 6     Velocidad Y: 5Color: 4
Particula numero 2: Posicion X: 511.372  Posicion Y: 209.647Velocidad X: 9     Velocidad Y: 2Color: 4
Particula numero 3: Posicion X: 545.336  Posicion Y: 31.3235Velocidad X: 6     Velocidad Y: 4Color: 7
Particula numero 4: Posicion X: 562.253  Posicion Y: 43.8737Velocidad X: 9     Velocidad Y: 4Color: 2
Particula numero 5: Posicion X: 582.443  Posicion Y: 314.239Velocidad X: 0     Velocidad Y: 5Color: 7
Particula numero 6: Posicion X: 511.372  Posicion Y: 209.647Velocidad X: 9     Velocidad Y: 2Color: 4
Particula numero 7: Posicion X: 545.336  Posicion Y: 31.3235Velocidad X: 6     Velocidad Y: 4Color: 7
Particula numero 8: Posicion X: 562.253  Posicion Y: 43.8737Velocidad X: 9     Velocidad Y: 4Color: 2
Particula numero 9: Posicion X: 582.443  Posicion Y: 314.239Velocidad X: 0     Velocidad Y: 5Color: 7


^C==8907==
==8907== Process terminating with default action of signal 2 (SIGINT)
==8907==    at 0x439DA6C: pthread_cond_wait@@GLIBC_2.3.2 (pthread_cond_wait.S:188)
==8907==    by 0x456E5EC: ??? (in /usr/lib/i386-linux-gnu/libxcb.so.1.1.0)
==8907==    by 0x456EBA4: ??? (in /usr/lib/i386-linux-gnu/libxcb.so.1.1.0)
==8907==    by 0x456EC5C: xcb_writev (in /usr/lib/i386-linux-gnu/libxcb.so.1.1.0)
==8907==    by 0x4094A51: _XSend (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==8907==    by 0x4094FE9: _XReply (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==8907==    by 0x40909AE: XSync (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==8907==    by 0x40712B0: XCloseDisplay (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==8907==    by 0x804C9E1: main (miniwin.cpp:740)
==8907==
==8907== HEAP SUMMARY:
==8907==     in use at exit: 61,579 bytes in 36 blocks
==8907==   total heap usage: 103 allocs, 67 frees, 71,016 bytes allocated
==8907==
==8907== 144 bytes in 1 blocks are possibly lost in loss record 23 of 31
==8907==    at 0x402F0B8: calloc (in /usr/lib/valgrind/vgpreload_memcheck-x86-linux.so)
==8907==    by 0x4011626: allocate_dtv (dl-tls.c:322)
==8907==    by 0x401204B: _dl_allocate_tls (dl-tls.c:539)
==8907==    by 0x4398CC3: allocate_stack (allocatestack.c:588)
==8907==    by 0x4398CC3: pthread_create@@GLIBC_2.1 (pthread_create.c:539)
==8907==    by 0x804C841: _maybe_call_main() (miniwin.cpp:673)
==8907==    by 0x804C89B: _process_event() (miniwin.cpp:689)
==8907==    by 0x804C996: main (miniwin.cpp:733)
==8907==
==8907== LEAK SUMMARY:
==8907==    definitely lost: 0 bytes in 0 blocks
==8907==    indirectly lost: 0 bytes in 0 blocks
==8907==      possibly lost: 144 bytes in 1 blocks
==8907==    still reachable: 61,435 bytes in 35 blocks
==8907==         suppressed: 0 bytes in 0 blocks
==8907== Reachable blocks (those to which a pointer was found) are not shown.
==8907== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==8907==
==8907== For counts of detected and suppressed errors, rerun with: -v
==8907== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)

jorgeklock@ei140182:~/NetBeansProjects/Practica6/dist/Debug/GNU-Linux$
```

Case 2:

Ésta es la salida de Valgrind para el caso 2 del main.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
==8963== Thread 2:
==8963== Syscall param writev(vector[...]) points to uninitialised byte(s)
==8963==    at 0x448E0BF: ??? (syscall-template.S:84)
==8963==    by 0x456E66D: ??? (in /usr/lib/i386-linux-gnu/libxcb.so.1.1.0)
==8963==    by 0x456EBA4: ??? (in /usr/lib/i386-linux-gnu/libxcb.so.1.1.0)
==8963==    by 0x456EC5C: xcb_writev (in /usr/lib/i386-linux-gnu/libxcb.so.1.1.0)
==8963==    by 0x4094A51: _XSend (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==8963==    by 0x4094DF8: _XFlush (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==8963==    by 0x4075CDF: XFlush (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==8963==    by 0x804CF2E: miniwin::refresca() (miniwin.cpp:845)
==8963==    by 0x804BA81: _main_() (main.cpp:98)
==8963==    by 0x804C809: _invoke_main(void*) (miniwin.cpp:665)
==8963==    by 0x4398294: start_thread (pthread_create.c:333)
==8963==    by 0x4495EED: clone (clone.S:114)
==8963==  Address 0x45ab6c4 is 644 bytes inside a block of size 16,384 alloc'd
==8963==    at 0x402F0B8: calloc (in /usr/lib/valgrind/vgpreload_memcheck-x86-linux.so)
==8963==    by 0x408476B: XOpenDisplay (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==8963==    by 0x804C5EC: _open_display() (miniwin.cpp:620)
==8963==    by 0x804C93F: main (miniwin.cpp:725)
==8963==
^C==8963==
==8963== Process terminating with default action of signal 2 (SIGINT)
==8963==    at 0x439DA6C: pthread_cond_wait@@GLIBC_2.3.2 (pthread_cond_wait.S:188)
==8963==    by 0x456E5EC: ??? (in /usr/lib/i386-linux-gnu/libxcb.so.1.1.0)
==8963==    by 0x456EBA4: ??? (in /usr/lib/i386-linux-gnu/libxcb.so.1.1.0)
==8963==    by 0x456EC5C: xcb_writev (in /usr/lib/i386-linux-gnu/libxcb.so.1.1.0)
==8963==    by 0x4094A51: _XSend (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==8963==    by 0x4094FE9: _XReply (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==8963==    by 0x40909AE: XSync (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==8963==    by 0x40712B0: XCloseDisplay (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==8963==    by 0x804C9E1: main (miniwin.cpp:740)
==8963==
==8963== HEAP SUMMARY:
==8963==     in use at exit: 61,579 bytes in 36 blocks
==8963==   total heap usage: 2,587 allocs, 2,551 frees, 841,348 bytes allocated
==8963==
==8963== Thread 1:
==8963== 144 bytes in 1 blocks are possibly lost in loss record 23 of 31
==8963==    at 0x402F0B8: calloc (in /usr/lib/valgrind/vgpreload_memcheck-x86-linux.so)
==8963==    by 0x4011626: allocate_dtv (dl-tls.c:322)
==8963==    by 0x401284B: _dl_allocate_tls (dl-tls.c:539)
==8963==    by 0x4398CC3: allocate_stack (allocatestack.c:588)
==8963==    by 0x4398CC3: pthread_create@@GLIBC_2.1 (pthread_create.c:539)
==8963==    by 0x804C841: _maybe_call_main() (miniwin.cpp:673)
==8963==    by 0x804C89B: _process_event() (miniwin.cpp:689)
==8963==    by 0x804C996: main (miniwin.cpp:733)
==8963==
==8963== LEAK SUMMARY:
==8963==    definitely lost: 0 bytes in 0 blocks
==8963==    indirectly lost: 0 bytes in 0 blocks
==8963==      possibly lost: 144 bytes in 1 blocks
==8963==    still reachable: 61,435 bytes in 35 blocks
==8963==         suppressed: 0 bytes in 0 blocks
==8963== Reachable blocks (those to which a pointer was found) are not shown.
==8963== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==8963==
==8963== For counts of detected and suppressed errors, rerun with: -v
==8963== Use --track-origins=yes to see where uninitialised values come from
==8963== ERROR SUMMARY: 201 errors from 2 contexts (suppressed: 0 from 0)
```

Case 3:

Ésta es la salida de Valgrind para el caso 3 del main.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
==8996==    by 0x4094DF8: _XFlush (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==8996==    by 0x4075CDF: XFlush (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==8996==    by 0x804CF2E: miniwin::refresca() (miniwin.cpp:845)
==8996==    by 0x804AACB: Simulador::Pintar(int) const (Simulador.cpp:32)
==8996==    by 0x804B8E5: _main_() (main.cpp:117)
==8996==    by 0x804C809: _invoke_main(void*) (miniwin.cpp:665)
==8996==    by 0x4398294: start_thread (pthread_create.c:333)
==8996==  Address 0x45aba0c is 1,484 bytes inside a block of size 16,384 alloc'd
==8996==    at 0x402F0B8: calloc (in /usr/lib/valgrind/vgpreload_memcheck-x86-linux.so)
==8996==    by 0x408476B: XOpenDisplay (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==8996==    by 0x804C5EC: _open_display() (miniwin.cpp:628)
==8996==    by 0x804C93F: main (miniwin.cpp:725)
==8996==
terminate called without an active exception
==8996==
==8996== Process terminating with default action of signal 6 (SIGABRT)
==8996==    at 0x43DAEA9: raise (raise.c:54)
==8996==    by 0x43DC406: abort (abort.c:89)
==8996==    by 0x421BD34: __gnu_cxx::__verbose_terminate_handler() (in /usr/lib/i386-linux-gnu/libstdc++.so.6.0.21)
==8996==    by 0x4219832: ??? (in /usr/lib/i386-linux-gnu/libstdc++.so.6.0.21)
==8996==    by 0x42198AC: std::terminate() (in /usr/lib/i386-linux-gnu/libstdc++.so.6.0.21)
==8996==    by 0x421932D: __gxx_personality_v0 (in /usr/lib/i386-linux-gnu/libstdc++.so.6.0.21)
==8996==    by 0x438A4A6: ??? (in /lib/i386-linux-gnu/libgcc_s.so.1)
==8996==    by 0x438A7AD: _Unwind_ForcedUnwind (in /lib/i386-linux-gnu/libgcc_s.so.1)
==8996==    by 0x43A2C29: _Unwind_ForcedUnwind (unwind-forcedunwind.c:134)
==8996==    by 0x43A0AD9: __pthread_unwind (unwind.c:121)
==8996==    by 0x44D364E: __pthread_unwind (forward.c:286)
==8996==    by 0x44A2A4A: __do_cancel (pthreadP.h:283)
==8996==    by 0x44A2A4A: __libc_enable_asynccancel (cancellation.c:49)
==8996==
==8996== HEAP SUMMARY:
==8996==     in use at exit: 63,395 bytes in 40 blocks
==8996==   total heap usage: 2,127 allocs, 2,087 frees, 123,348 bytes allocated
==8996==
==8996== Thread 1:
==8996== 144 bytes in 1 blocks are possibly lost in loss record 23 of 35
==8996==    at 0x402F0B8: calloc (in /usr/lib/valgrind/vgpreload_memcheck-x86-linux.so)
==8996==    by 0x4011626: allocate_dtv (dl-tls.c:322)
==8996==    by 0x401204B: _dl_allocate_tls (dl-tls.c:539)
==8996==    by 0x4398CC3: allocate_stack (allocatestack.c:588)
==8996==    by 0x4398CC3: pthread_create@@GLIBC_2.1 (pthread_create.c:539)
==8996==    by 0x804C841: _maybe_call_main() (miniwin.cpp:673)
==8996==    by 0x804C89B: _process_event() (miniwin.cpp:689)
==8996==    by 0x804C996: main (miniwin.cpp:733)
==8996==
==8996== LEAK SUMMARY:
==8996==    definitely lost: 0 bytes in 0 blocks
==8996==    indirectly lost: 0 bytes in 0 blocks
==8996==      possibly lost: 144 bytes in 1 blocks
==8996==    still reachable: 63,251 bytes in 39 blocks
==8996==                       of which reachable via heuristic:
==8996==                         newarray          : 1,816 bytes in 4 blocks
==8996==         suppressed: 0 bytes in 0 blocks
==8996== Reachable blocks (those to which a pointer was found) are not shown.
==8996== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==8996==
==8996== For counts of detected and suppressed errors, rerun with: -v
==8996== Use --track-origins=yes to see where uninitialised values come from
==8996== ERROR SUMMARY: 504 errors from 2 contexts (suppressed: 0 from 0)
```

Case 4:

El main proporcionado para la práctica constaba de 3 opciones, ninguna de las cuales se repetía de forma infinita hasta que el usuario lo decidiese. Eso quitaba la opción de probar la segunda Ampliación que había realizado.

Por ésto, añadí una cuarta opción análoga al main que teníamos para la parte B del Proyecto, en el que se ejecuta la simulación mientras no se pulse ESCAPE y en dicha opción añadí la opción de reiniciar usando ESPACIO. (Fig. 3)

```
case 4:
{
    int ancho = 600 , alto = 400;
    srand (time (0)) ;
        ConjuntoParticulas parts (15) ;
        ConjuntoParticulas obstaculos (5) ;
        ConjuntoParticulas aux ;
        // PONER VELOCIDADES OBSTACULOS A 0
        for(int i=0; i<5; i++)
        {
            obstaculos.ObtieneParticulaReal(i)->setdx(0);
            obstaculos.ObtieneParticulaReal(i)->setdy(0);
        }
            Simulador mySim (parts ,obstaculos ,ancho ,alto ) ;
            int t= tecla();
                while (t != ESCAPE)
                {
                mySim.Step () ;
                mySim.Pintar (25) ;
                    if(mySim.getFijas().getutiles()<20)
                    {
                        aux = mySim.getFijas() ;
                        Particula p;
                        aux.AgregaParticula(p);
                        mySim.setFijas(aux);
                    }
                    if(t == ESPACIO)
                    {
                    mySim.setMoviles(parts);
                    mySim.setFijas(obstaculos);
                    }
                t=tecla();
                }
}
```

Fig. 3

Escribiré la autoevaluación al pie de esta página y a continuación la salida del programa Valgrind para ésta cuarta opción del main ya que al ser tan largo necesita mucho espacio de página para poder leerse decentemente.

Auto-Evaluación: Sobresaliente

Archivo Editar Ver Buscar Terminal Ayuda

```
==9028==    by 0x4075CDF: XFlush (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==9028==    by 0x804CF2E: miniwin::refresca() (miniwin.cpp:845)
==9028==    by 0x804AACB: Simulador::Pintar(int) const (Simulador.cpp:32)
==9028==    by 0x804BDCA: _main_() (main.cpp:142)
==9028==    by 0x804C889: _invoke_main(void*) (miniwin.cpp:665)
==9028==    by 0x4398294: start_thread (pthread_create.c:333)
==9028==  Address 0x45ab79c is 860 bytes inside a block of size 16,384 alloc'd
==9028==    at 0x402F0B8: calloc (in /usr/lib/valgrind/vgpreload_memcheck-x86-linux.so)
==9028==    by 0x408476B: XOpenDisplay (in /usr/lib/i386-linux-gnu/libX11.so.6.3.0)
==9028==    by 0x804C5EC: _open_display() (miniwin.cpp:620)
==9028==    by 0x804C93F: main (miniwin.cpp:725)
==9028==
terminate called without an active exception
==9028==
==9028== Process terminating with default action of signal 6 (SIGABRT)
==9028==    at 0x43DAEA9: raise (raise.c:54)
==9028==    by 0x43DC406: abort (abort.c:89)
==9028==    by 0x421BD34: __gnu_cxx::__verbose_terminate_handler() (in /usr/lib/i386-linux-gnu/libstdc++.so.6.0.21)
==9028==    by 0x4219832: ??? (in /usr/lib/i386-linux-gnu/libstdc++.so.6.0.21)
==9028==    by 0x42198AC: std::terminate() (in /usr/lib/i386-linux-gnu/libstdc++.so.6.0.21)
==9028==    by 0x421932D: __gxx_personality_v0 (in /usr/lib/i386-linux-gnu/libstdc++.so.6.0.21)
==9028==    by 0x438A4A6: ??? (in /lib/i386-linux-gnu/libgcc_s.so.1)
==9028==    by 0x438A7AD: _Unwind_ForcedUnwind (in /lib/i386-linux-gnu/libgcc_s.so.1)
==9028==    by 0x43A2C29: _Unwind_ForcedUnwind (unwind-forcedunwind.c:134)
==9028==    by 0x43A0AD9: __pthread_unwind (unwind.c:121)
==9028==    by 0x44D364E: __pthread_unwind (forward.c:206)
==9028==    by 0x44A2A4A: __do_cancel (pthreadP.h:283)
==9028==    by 0x44A2A4A: __libc_enable_asynccancel (cancellation.c:49)
==9028==
==9028== HEAP SUMMARY:
==9028==     in use at exit: 84,227 bytes in 76 blocks
==9028==   total heap usage: 4,296 allocs, 4,220 frees, 1,730,912 bytes allocated
==9028==
==9028== Thread 1:
==9028== 144 bytes in 1 blocks are possibly lost in loss record 27 of 44
==9028==    at 0x402F0B8: calloc (in /usr/lib/valgrind/vgpreload_memcheck-x86-linux.so)
==9028==    by 0x4011626: allocate_dtv (dl-tls.c:322)
==9028==    by 0x401204B: _dl_allocate_tls (dl-tls.c:539)
==9028==    by 0x4398CC3: allocate_stack (allocatestack.c:588)
==9028==    by 0x4398CC3: pthread_create@@GLIBC_2.1 (pthread_create.c:539)
==9028==    by 0x804C841: _maybe_call_main() (miniwin.cpp:673)
==9028==    by 0x804C89B: _process_event() (miniwin.cpp:689)
==9028==    by 0x804C996: main (miniwin.cpp:733)
==9028==
==9028== LEAK SUMMARY:
==9028==    definitely lost: 0 bytes in 0 blocks
==9028==    indirectly lost: 0 bytes in 0 blocks
==9028==      possibly lost: 144 bytes in 1 blocks
==9028==    still reachable: 84,083 bytes in 75 blocks
==9028==                       of which reachable via heuristic:
==9028==                         newarray         : 15,620 bytes in 5 blocks
==9028==         suppressed: 0 bytes in 0 blocks
==9028== Reachable blocks (those to which a pointer was found) are not shown.
==9028== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==9028==
==9028== For counts of detected and suppressed errors, rerun with: -v
==9028== Use --track-origins=yes to see where uninitialised values come from
==9028== ERROR SUMMARY: 616 errors from 2 contexts (suppressed: 0 from 0)
Terminado (killed)
```