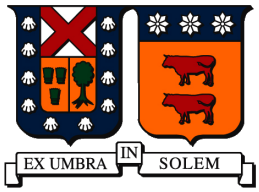


UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

DEPARTAMENTO DE ELECTRÓNICA

Diseño y Programación Orientados a Objetos

Tarea N^{o}3
"Bolas Libres y Enlazadas en Espacio Cerrado como Objeto de Software en C++"

02 de Junio del 2016

GRUPO

Jesús Márquez 201221009-7
Diego Pandolfa 201221069-0

Índice

Índice de Imágenes

Índice de Tablas

1 Etapa 1.

Se simulará el choque de dos bolas sin roce en un espacio sin contenedores. Para esto se utilizaron los siguientes parámetros:

	Posición (x , y) [m]	Velocidad (x , y) [m/s]	Masa [kg]	Radio [m]
Bola 1	(1,0.5)	(0.5,0.5)	1	0.1
Bola 2	(2,1.5)	(-0.5,-0.5)	1	0.1

Tabla 1: Parámetros de la configuración simulada.

En la figura a continuación se muestran las trayectorias de ambas pelotas con las condiciones antes expuestas.

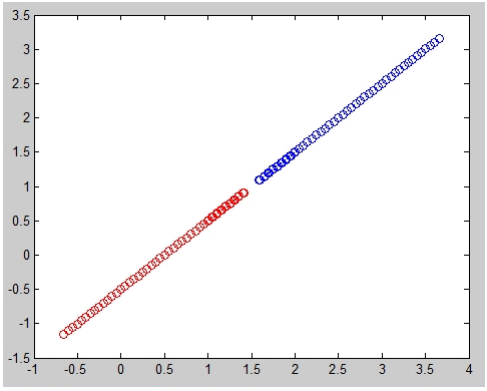
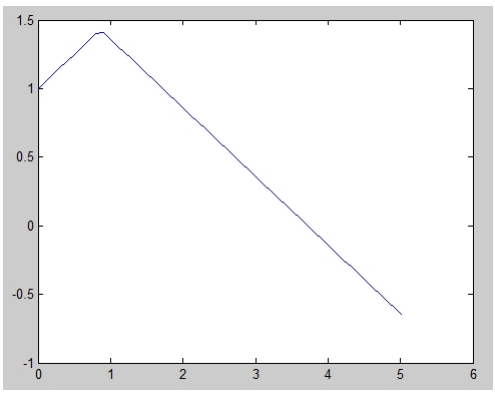
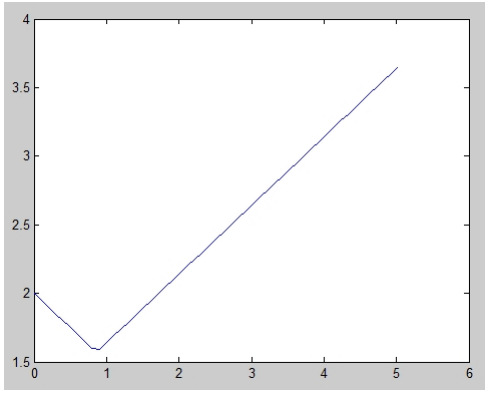


Figura 1: Trayectoria Y vs X de las bolas en la etapa 1



(a) Bola 0: Posicion X vs Tiempo



(b) Bola 1: Posicion X vs Tiempo

Figura 2: *Posición X de las bolas a través del tiempo*

Se puede observar claramente que el choqe ocurre cercano al tiempo 1 [s].

1.1 Dificultades y/o Observaciones.

En la primera etapa ocurren varias complicaciones, la primera de ellas es que existe una mala ubicación de un else, colocandose en un ciclo más interno al deseado, lo que provocaba que la velocidad de las bolas se setee en cero, por lo que las bolas quedaban estaticas. El problema de esto es que tomo más tiempo del debido dar con él, por lo que generó un retraso mayor al esperado.

Además, existe un mal entendimiento del método collideWith(Ball*), ya que al implementarlo por primera vez, si la colisión ocurría, se retornaba un puntero al argumento del metodo, siendo que lo esperado era retornar un puntero a la instancia que se le aplicaba el método. Esto conlleva a que al momento de calcular la diferencia de posiciones, se obtuviera un vector nulo y luego, al dividir un numero por el modulo de dicho vector, provoque un error y retorne nan (Not a number).

2 Etapa 2.

Para esta etapa, se pide la colisión entre bolas pero teniendo un contenedor del que no puedan escapar. Se hizo una simulación con 2 bolas y un contenedor en donde se usaron los siguientes parámetros:

	Posición (x , y) [m]	Velocidad (x , y) [m/s]	Masa [kg]	Radio [m]
Bola 1	(1,1)	(0.5,0.5)	1	0.1
Bola 2	(2,2)	(-0.5,-0.5)	1	0.1

Tabla 2: Parámetros de la configuración simulada.

En la figura a continuación se muestran las trayectorias de ambas pelotas con las condiciones antes expuestas.

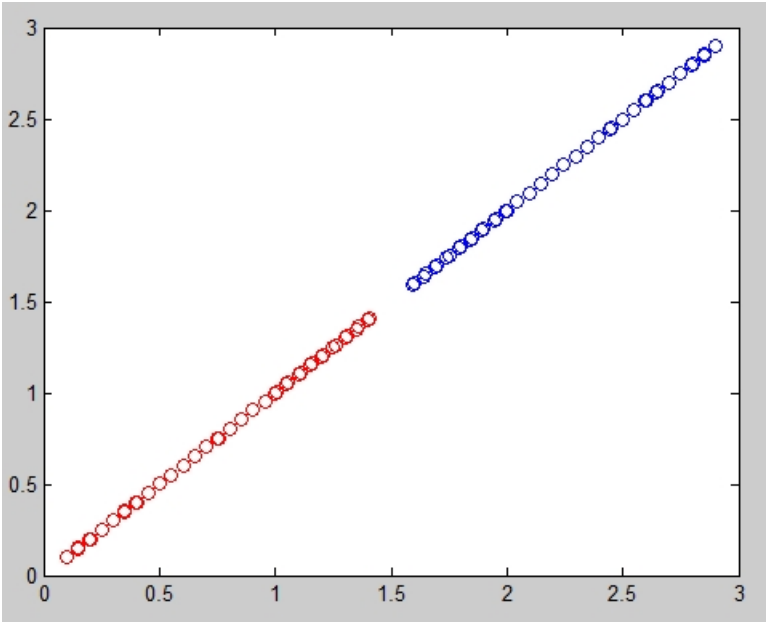
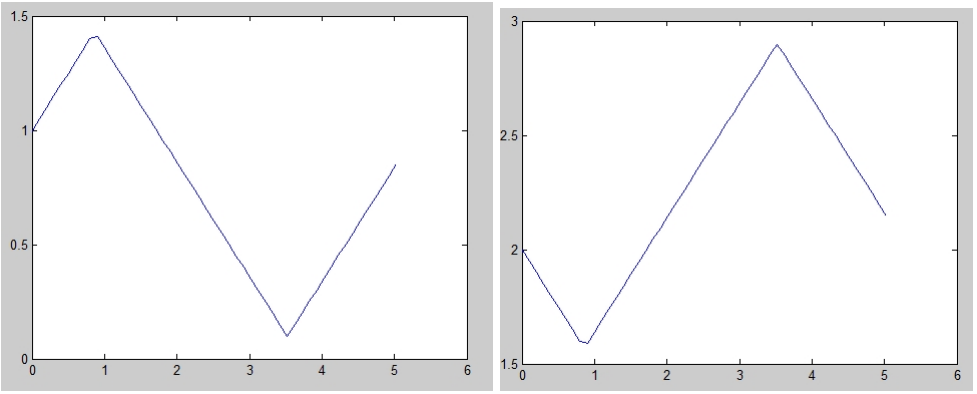


Figura 3: Trayectoria Y vs X de las bolas en la etapa 2



(a) Bola 0: Posicion X vs Tiempo (b) Bola 1: Posicion X vs Tiempo

Figura 4: *Posición X de las bolas a través del tiempo*

Se puede observar claramente que los choqes ocurren en los instantes cercanoa al tiempo 1 [s] y 3.5 [s].

2.1 Dificultades y/o Observaciones.

Al diseñar la etapa 2 ocurren varios errores, el primero de ellos ocurre con la clase Container, ya que cuando se detecta que hay una colisión, se llama a un metodo que retorna un puntero a un arreglo de murallas. Sin embargo, al recorrerlo, se obtienen datos basura, lo que implica que al calcular la velocidad de la bola, se obtienen mas datos basura, por lo que la posición termina siendo nan. Esto se debe a que al llamar el metodo que entregaba el puntero, este retornaba un arreglo con murallas hechas ahi mismo, por lo que al terminar de ejecutarse, se eliminaban y los punteros terminan apuntando a nada concreto.

Luego, al intentar reparar ese problema, se intenta inicializar dicho arreglo en el constructor de la clase Container, sin embargo el compilador no permitia realizar la asignación, ya que señalaba que los tipos no coincidian, lo que no se explica ya que se llama al constructor diseñado y probado en otras partes del codigo. Este error no se logra solucionar, por lo que se opta a utilizar un vector de murallas, el cual es rellenado en el constructor y despues, para entregar las murallas colindantes (pueden ser 2 al chocar en una esquina), se recorre éste y se asigna a otro vector las murallas implicadas, el cual es retornado como resultado del metodo.

Finalmente, se presenta un error al verificar la condición del choque de las murallas, ya que al efectuar el calculo que permite corroborar que la muralla toca a la bola, se utilizaba la posición de la bola con respecto al origen, lo que generaba que si habia murallas que eran paralelas entre si, si la bola tocaba a alguna de ellas, el algoritmo señalaba que tocaba a ambas murallas y "chocaba" dos veces, por lo que cambiaba su velocidad 2 veces, quedando igual que al inicio. Esto se corrige al comprender mejor los apuntes y utilizar la posición de la bola, pero respecto al origen de la muralla, no al punto 0,0.

3 Parte 3.

Para esta etapa se conectaran dos bolas a un resorte y se dejaran oscilar en un ambiente cerrado por un container, se hara una prueba con los siguientes parámetros:

	Posición (x , y) [m]	Velocidad (x , y) [m/s]	Masa [kg]	Radio [m]
Bola 0	(-2,-2)	(-1,0)	1	0.1
Bola 1	(2,2)	(-1,0)	1	0.1

Tabla 3: Parámetros de la configuración simulada.

En la figura a continuación se muestran las trayectorias de ambas pelotas con las condiciones antes expuestas.

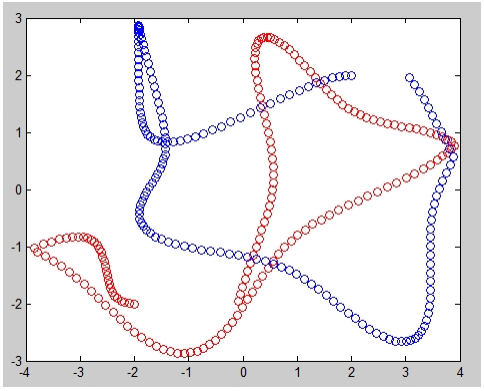
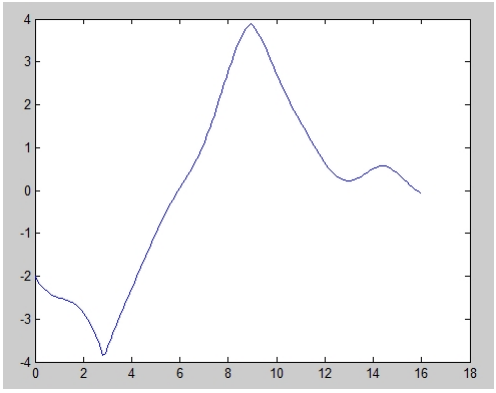
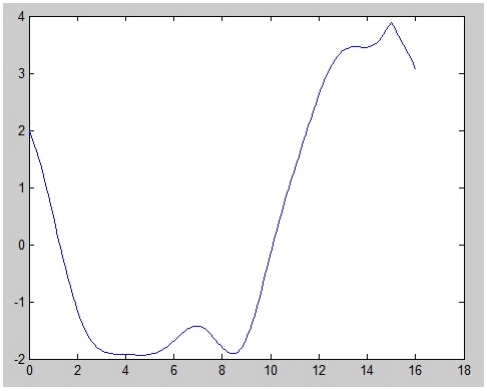


Figura 5: Trayectoria Y vs X de las bolas en la etapa 3



(a) Bola 0: Posicion X vs Tiempo



(b) Bola 1: Posicion X vs Tiempo

Figura 6: *Posición X de las bolas a través del tiempo*

Se puede observar claramente que los choqes ocurren en los instantes cercanoa al tiempo 3 [s] , 4 [s] , 9 [s] y 15 [s].

3.1 Dificultades y/o Observaciones.

La mayor dificultad que tuvimos fue que la sobrecarga de operadores estaba mal implementada y no se calculaban bien los siguientes estados.

4 Parte 4.

Para esta etapa se conectaron más de un resorte a cada bola de tal forma que estan todas conectadas, es decir, la Bola 0 esta conectada a la Bola 1 y 2, de la misma forma la Bola 1 esta conectada con las Bolas 0 y 2, como también la Bola 2 está conectada con las Bolas 0 y 1. Además se encuentran encerradas en un contenedor de [-4,4]x[-4,4]. A continuación se muestra una tabla con los parámetros de la simulación.

	Posición (x , y) [m]	Velocidad (x , y) [m/s]	Masa [kg]	Radio [m]
Bola 0	(-2,-2)	(2,0)	1	0.1
Bola 1	(2,2)	(2,0)	1	0.1
Bola 2	(2,-2)	(2,0)	1	0.1

Tabla 4: Parámetros de la configuración simulada.

En la figura a continuación se muestran las trayectorias de las 3 pelotas con las condiciones antes expuestas.

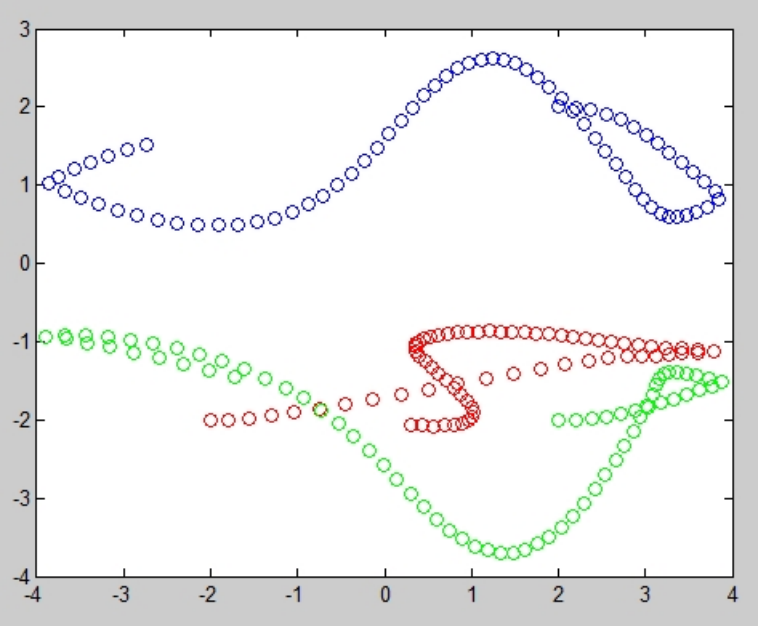
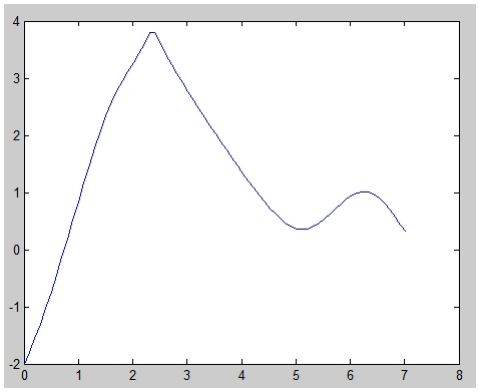
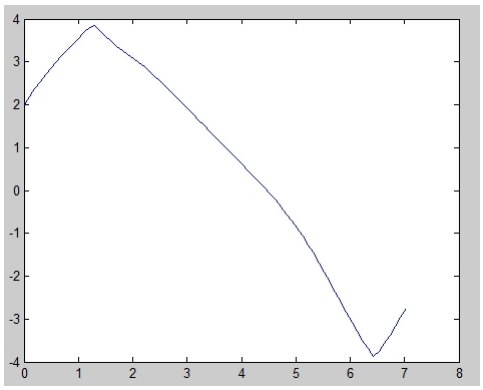


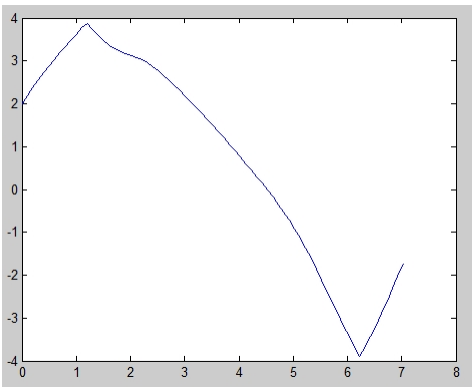
Figura 7: Trayectoria Y vs X de las bolas en la etapa 4



(a) Bola 0: Posicion X vs Tiempo



(b) Bola 1: Posicion X vs Tiempo



(c) Bola 2: Posicion X vs Tiempo

Figura 8: *Posición X de las bolas a través del tiempo*

Se puede observar claramente que los choqes ocurren en los instantes cercanoa al tiempo 1 [s] , 2 [s] , 6 [s] y 6.5 [s].

4.1 Dificultades y/o Observaciones.

No se tuvieron mayores dificultades en esta etapa, ya que escencialmente era lo mismo que la etapa anterior.

4.2 Diagrama de Clases

A continuación se muestra el diagrama de clases entregado por el software Visual Studio 2015 Enterprice:

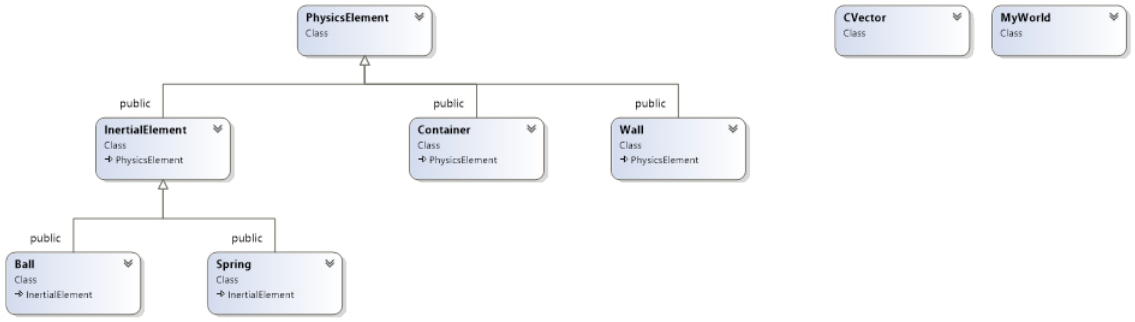


Figura 9: Diagrama de Clases.