

Universidad Tecnica Federico Santa Maria

Departamento de Electrónica

ELO329 - Diseño y Programación Orientados a Objetos

1er Semestre 2016

Documentación

Tarea 1

Tomas Gomez Molina

Johannes Rothkegel Sielfeld

Paralelo 1

14 de mayo de 2016

1. Etapa 1

En esta etapa solo fue necesario seguir los pasos de la actividad en clases, en conjunto con el profesor.

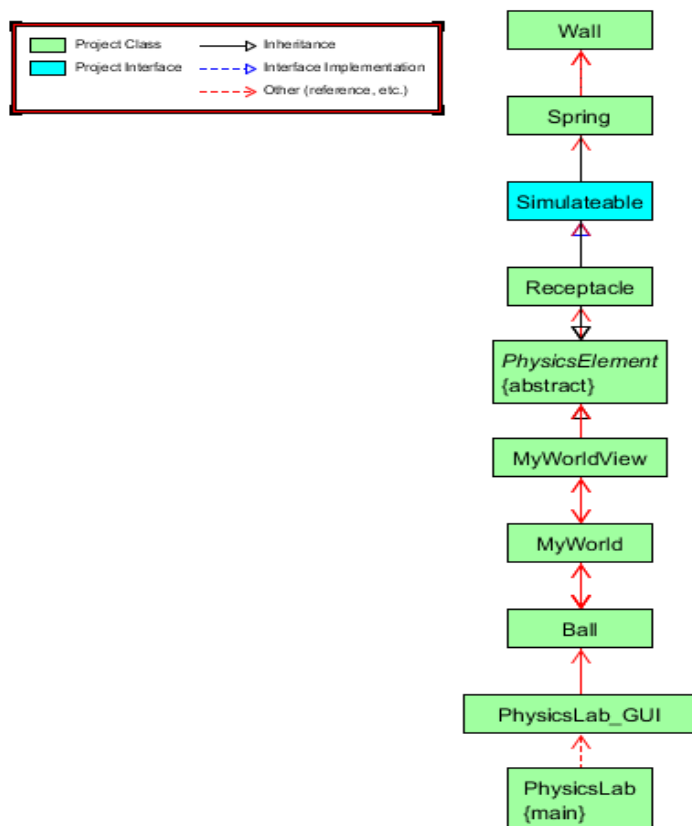


Figura 1: Diagrama de Clases para la etapa 1, excluyendo la clases Vector2D y las terminadas en View.

1.1. Dificultades y/o Observaciones

Las dificultades en ésta etapa fueron mínimas, lo más complejo fue arreglar los errores de compilación y dibujar el contenedor. Lo demás no generó mayor problema. Un comentario es que luego de varias pruebas se comprobó que si una bola choca exactamente en 45° con el exterior de un contenedor, éste permite el ingreso de la bola como si no hubiera una colisión.

2. Etapa 2

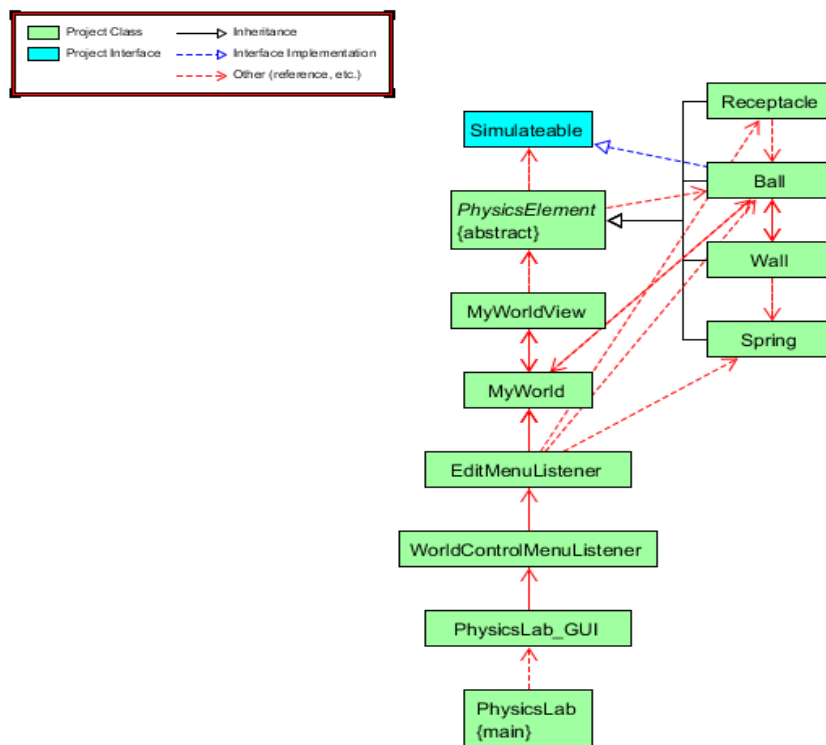


Figura 2: Diagrama de Clases para la etapa 2, excluyendo la clases Vector2D y las terminadas en View.

Para la etapa 2, se modificaron las clases **WorldControlMenuListener** y **PhysicsLab** para visualizar el menu tal como se pide en la tarea. Luego se modifco la clase **EditMenuListener** para poder ingresar el spring a la parte visual. Para esto ultimo se inicializo un spring con dos bolas invisibles para el mundo, las cuales sirven solamente como coordenadas para los extremos del spring.

2.1. Dificultades y/o Observaciones

Las dificultades que más complicaron ésta etapa fue comprender el sistema modelo-vista-controlador, para luego aplicarlo en el programa, con los listeners del menú y las modificaciones que estos eventos tenían en los objetos.

3. Etapa 3

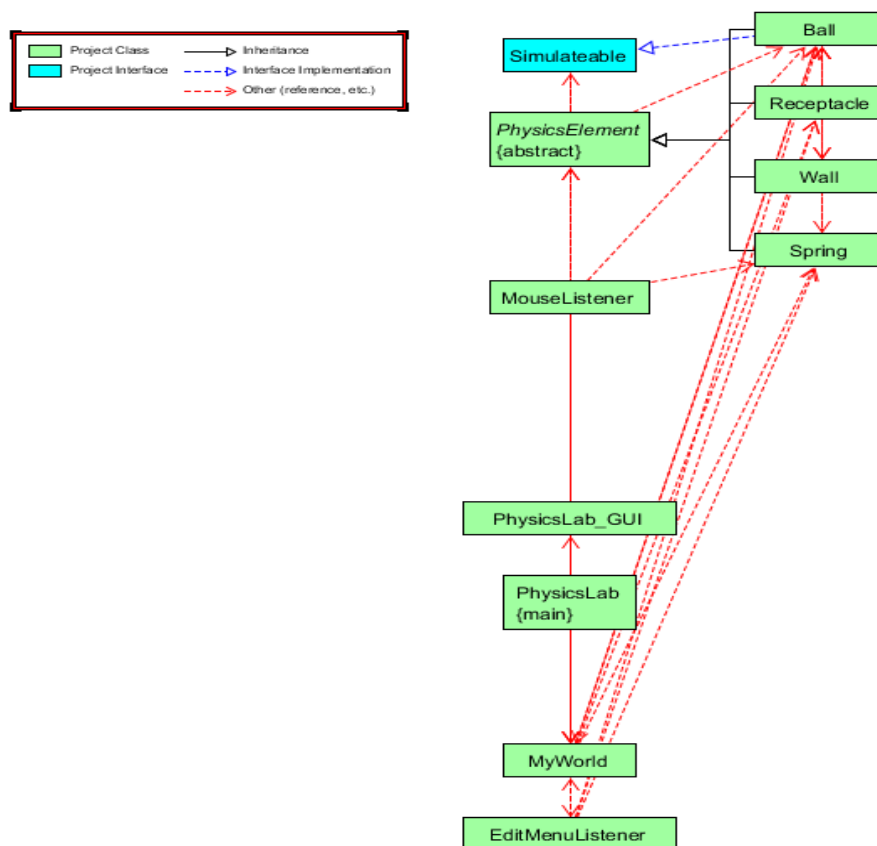


Figura 3: Diagrama de Clases para la etapa 2, excluyendo la clases Vector2D y las terminadas en View.

En esta etapa se modificaron las clases Spring, MyWorld, MouseListener y PhysicsElement.

En la clase PhysicsElement se creo un nuevo metodo abstracto el cual es funcional solo para la clase spring, este se llama dragbDelta. Este fue creado con el proposito de mover el extremo ligado a bBall.

A las clases View se les agregaron los metodos isSelected() e isReleased().

Luego a la clase Spring se le modificaron los metodos de forma que cumplan los requisitos pedidos.

La clase MyWorld fue trabajada minuciosamente. En esta se creo el metodo findElement de modo que se pueda ver si en esa posicion existe un elemento.

Finalmente en la clase MouseListener se modificaron los metodos acorde a las especificaciones

pedidas en la tarea. El metodo `mouseReleased()` describe detalladamente para el caso que se quiera aderir un sping a una bola.

3.1. Dificultades y/o Observaciones

En esta etapa el mayor problema fue el resorte de la clase `Spring`, pues la configuración de la unión del resorte con la bola que se encontrara en el punto donde se soltó el resorte fue muy complicado, se dejó un pequeño detalle que es que al agregar un spring antes que una bola, el spring no será capaz de unirse a esa bola, el error fue tratado bastante tiempo sin lograr corrección, una idea fue modificar la lista de elementos para que el resorte quedara antes que las bolas al momento de ser seleccionado, por cuestión de tiempo no se arregló. Otro detalle es que para liberar al resorte de la bola se debe hacer la selección del resorte por el extremo que queda libre.

4. Etapa 4

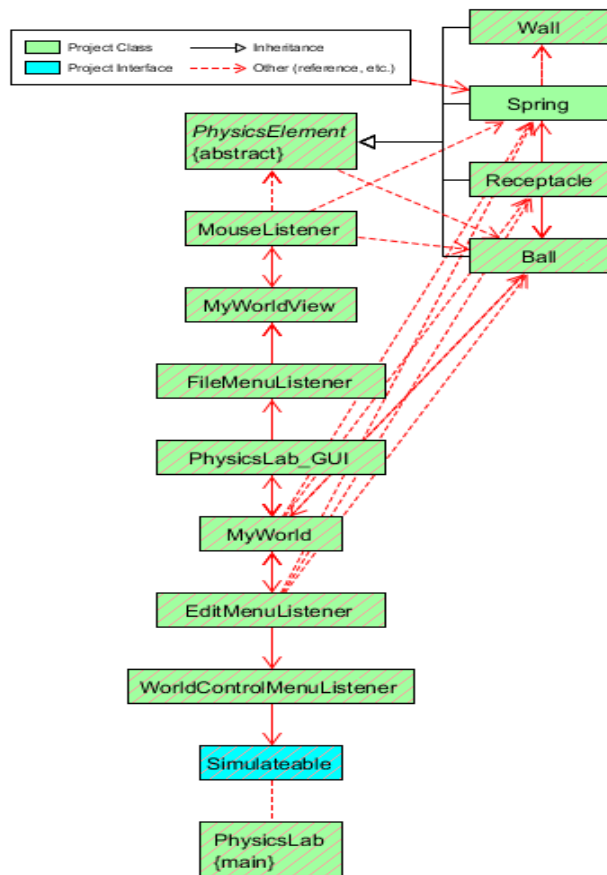


Figura 4: Diagrama de Clases para la etapa 2, excluyendo la clases Vector2D y las terminadas en View.

4.1. Dificultades y/o Observaciones

En la última etapa la única modificación realizada fue agregar el listener **FileMenuListener** que se encargaba de agregar una barra más en el menú y permitirle al usuario guardar o cargar los datos de la simulación, algunas dificultades fueron que luego de que funcionara perfectamente, lanzaba algunas notas en el compilador, después de investigar en internet se decidió ignorar los mensajes ya que solo advertían la manera con la que se estaba trabajando con la clase **io** de java.