

Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Ingeniería en Computación

“Simulador cuerpos rígidos: esfera y terreno accidentado”

Alumno:

Julio César Rangel Reyes

Matricula: 204308656

Trimestre 11-I

Asesor:

Dr. Risto Rangel Kuoppa

CONTENIDO

Introducción.....	3
Justificación.....	4
Objetivo General.....	4
Objetivos Particulares.....	5
Desarrollo.....	5
Detección de colisiones.....	5
implementación de leyes las de newton.....	6
La primera ley.....	7
La segunda ley.....	7
Fuerza de gravedad.....	7
actualización de la posición y velocidad de la esfera.....	8
Reaccionando a las colisiones (tercera ley de newton).....	8
Dirección de la colisión	9
Manual de la aplicación.....	10
Ventana de visualización:.....	10
La ventana de parámetros de la esfera.....	11
Cargando otros terrenos.....	11
Diagrama de clases.....	12
Diagramas de Entidad-Relación.....	13
.....	13
Diagrama Estados.....	14
DIAGRAMA DE OBJETOS.....	15
DIAGRAMA DE FLUJO.....	16
Bibliografía:.....	17

INTRODUCCIÓN

Este reporte final refleja el desarrollo del trabajo que se llevo a cabo en la realización de mí de proyecto terminal en Ingeniería en computación, también contiene el manual de usuario del software realizado en este mismo.

La primera parte del proyecto fue diseñar e implementar una interfaz gráfica que fuera capaz de:

- Renderizar¹ una esfera y un terreno² accidentado en un ambiente virtual tridimensional.
- Modificar la masa, posición y coeficiente de sustitución de la esfera³.
- Modificar la posición de la cámara⁴.
- Cargar terrenos pre-creados de un archivo.

Después de haber construido una buena base para visualizar nuestros cuerpos rígidos, se prosiguió a la implementación de un algoritmo sencillo para detectar colisiones entre la esfera y el terreno, cabe mencionar que este algoritmo no era el más eficiente, pero sí el más intuitivo de implementar. Cuando el algoritmo funcionó satisfactoriamente, se prosiguió con la parte más interesante del proyecto(a mí parecer), la cual fue hacer que la esfera simulara estar bajo las leyes del movimiento de Newton.

La primera ley la cuál trata del estado en reposo de un objeto mientras no se le ejerza una fuerza externa a éste, esto simplemente se dio por implementado dado que nuestra esfera al no implementar fuerzas a esas alturas cumplía dicha ley.

Para implementar la segunda y tercera ley, se tuvo que investigar en un libro de física[1] el marco teórico de estas y extraer las ecuaciones de movimiento de dichas leyes. Después de esto se cayó en la necesidad de implementar y diseñar las abstracciones necesarias en clases para hacer experimentar a la esfera fuerzas como gravedad, y las relacionadas con colisiones.

Por último en la fase pruebas, se notó que el algoritmo implementado para detectar las colisiones, era muy lento, así que optó por implementar otro más eficiente, más adelante se detalla.

¹ Es la acción de asignar y calcular todas las propiedades de un objeto antes de mostrarlo en pantalla

² Terreno accidentado

³ Coeficiente de restitución: Sirve para quitar energía a un cuerpo después de una colisión.

⁴

JUSTIFICACIÓN

La simulación por computadora se ha convertido en una parte útil del modelado de muchos sistemas naturales en física, astrofísica, química y biología, y sistemas humanos como la economía y las ciencias sociales (sociología computacional).

En general, la simulación por computadora tiene buenas aplicaciones donde el modelado matemático por sí sólo no basta o donde se vuelve imposible de hacer dado el número de variables, un ejemplo es la simulación de la colisión de dos galaxias, donde se simulan periodos de millones de años de interacciones de estrellas y otros cuerpos celestes, a partir de haber sucedido dicha colisión.



Simulación de la colisión de dos galaxias, en esa foto a pasado 1 billón de años.
<http://hubblesite.org/newscenter/archive/releases/2008/16/video/d/>

En resumen existe la necesidad de simular fenómenos físicos sin tener que recrearlos en la realidad, por qué esto exigiría un gasto muy grande de recursos. Otra razón podría ser que simplemente no es factible llevarlos a cabo por qué no existen los materiales (por ejemplo simular la caída de una canica de 5 cm de diámetro y masa de una tonelada).

Otro campo además del científico donde se usan motores de física, es en los videojuegos donde sirven para dar mayor realismo y mejorar la experiencia del usuario. Actualmente la mayoría de los videojuegos exitosos tratan de llevar al usuario a experimentar sensaciones, mediante un videojuego que simule fenómenos físicos de la vida real.

OBJETIVO GENERAL

Implementar un software que simule de forma realista⁵ una esfera sujeta a las 3 leyes de movimiento de newton en un terreno accidentado, ambos indeformables en un entorno virtual tridimensional.

⁵ Realista en el sentido que se vea muy parecido a lo que pasaría en una colisión en la realidad. No realista en el detalle de los objetos.

OBJETIVOS PARTICULARES

- Diseñar la interfaz gráfica que permita modificar la posición y masa de la esfera, así como desplegar los valores de la velocidad, aceleración y posición.
- Diseñar e implementar un esquema de visualización que despliegue el ambiente 3D simulado en tiempo real y que refleje la velocidad, aceleración y posición de la esfera conforme al tiempo.
- Implementar un modulo para cargar terrenos pre-creados al simulador desde un archivo.
- Analizar y diseñar un esquema de simulación de cuerpos rígidos, donde la esfera tenga las propiedades físicas de: posición, velocidad, aceleración, masa, rigidez; así como, que el terreno accidentado tenga la propiedad de rigidez.
- Analizar, diseñar e implementar el algoritmo más adecuado para generar un efecto de gravedad en la esfera.
- Analizar, diseñar e implementar el algoritmo más adecuado para detectar con rapidez las colisiones entre la esfera y el suelo.
- Analizar, diseñar e implementar el algoritmo más adecuado para responder con exactitud a las colisiones del terreno y la esfera.

DESARROLLO

DETECCIÓN DE COLISIONES

Un detector de colisiones⁶ es un código dedicado a buscar pares de objetos que estén colisionando, u objetos que este chocando con un pedazo de escenario no movable, como el terreno es nuestro caso.

En nuestro detector de colisiones el resultado final es el índice del primer vértice del triangulo del terreno donde colisiona con la esfera, o más específicamente donde el radio es menor que las distancias a sus vértices. Una vez que se tiene ese triangulo se puede calcular la normal de este para responder a la colisión.

Esta fue la parte que más demandó el uso del procesador, por lo que se tuvieron que hacer varias pruebas con dos algoritmos para ver cuál daba la combinación de realismo y eficiencia requeridos.

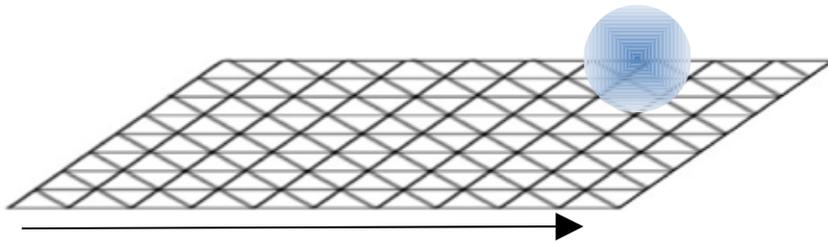
El primer algoritmo fue:

```
Desde primer triangulo hasta el último triangulo
Inicia
  Si (la distancia cada vértices <= radio de la esfera)
    Inicia
      // Hubo una colisión
```

⁶ Una colisión aquí, es cuando dos objetos simplemente se están tocando, aunque en la vida real suene como algo violento que pasa a gran velocidad.

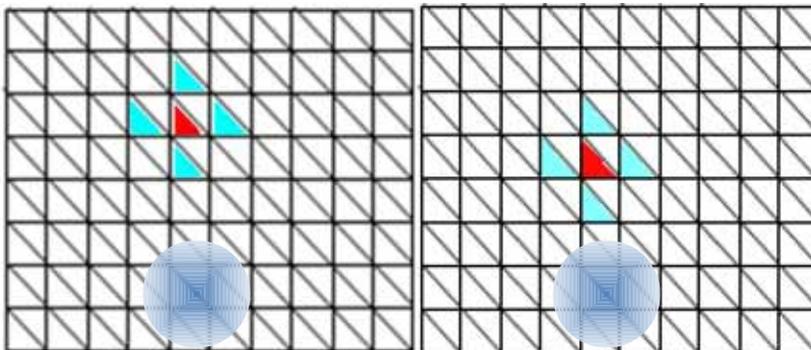
```
// Aplicar impulsos.  
Fin  
Fin
```

Este algoritmo recorre cada triángulo desde el primero de abajo a la izquierda, hasta el último de arriba a la derecha:



Dado que el algoritmo anterior era muy lento, se implementó otro, sugerido por el asesor de proyecto.

Este básicamente, se sitúa en un triángulo aleatorio (en rojo en la figura), se mira los triángulos cercanos (azules):



Entonces se busca un triángulo que esté más cerca del centro de la esfera que el triángulo actual (rojo), si existe, el triángulo actual se cambia a este más cercano.

Cuando ya no encuentra un triángulo más cercano, este comprueba si hay colisión.

IMPLEMENTACIÓN DE LEYES LAS DE NEWTON

Newton creó tres leyes de movimiento que describen con gran precisión cómo se comportan cuerpos con masa. En este proyecto nos enfocamos al comportamiento de esferas rígidas, las cuales dada la naturaleza

de este proyecto en el que no hay fricciones ni rotaciones, tienen posición pero no orientación, es decir, no sabemos a dónde apunta la esfera, o más bien no tiene sentido. Cabe mencionar que los resultados siguientes son para una esfera y terrenos indeformables, además de un terreno de masa infinita, es decir que no se mueve para nada.

Para las esferas de este proyecto se necesitó guardar varias propiedades: posición actual, velocidad, y aceleración. Todos estos son vectores, vectores de tres dimensiones.

LA PRIMERA LEY

La primera ley nos dice que pasa si no hay fuerzas alrededor, el objeto continuará moviéndose con velocidad constante. En nuestra aplicación asumiremos que se cumple la ley de Newton, dado que nuestra esfera está estática mientras no le apliquemos una fuerza.

LA SEGUNDA LEY

La segunda ley nos da los mecanismos con los que las fuerzas afectan el movimiento de un objeto. Una fuerza es algo que cambia la aceleración⁷ de un objeto.

FUERZA DE GRAVEDAD

La fuerza de gravedad es la fuerza más importante de la aplicación. La gravedad se aplica entre cada par de objetos atrayéndolos con una fuerza que depende de su masa, y de la distancia de sus centros.

$$f = G \frac{(m_1 m_2)}{r^2}$$

$$f = ma$$

$$a = G \frac{m_{tierra}}{r^2}$$

La distancia del centro de la tierra a la superficie (r) es de 6400 km aproximadamente, por lo tanto casi no hay diferencia entre la gravedad de estar al nivel del mar que estar de pie en una montaña, para la precisión requerida se puede tener r constante.

La constante a , es el valor entonces con el que cada cuerpo se acelera hacia el centro de la tierra. Tiene un valor aproximado de 9.8 m/s. En nuestra aplicación la gravedad actúa en dirección del suelo. La aceleración debido a la gravedad se puede representar como un vector de esta forma:

$$g = \begin{bmatrix} 0 \\ -a \\ 0 \end{bmatrix}$$

Donde g es el valor discutido anteriormente.

⁷ El ritmo de cambio de la velocidad.

ACTUALIZACIÓN DE LA POSICIÓN Y VELOCIDAD DE LA ESFERA

La actualización de la esfera se lleva a cabo en dos fases, la actualización de la posición y la actualización de la velocidad. La posición se actualiza por la bien conocida formula:

$$\mathbf{p}' = \mathbf{p} + \mathbf{v}t + \frac{1}{2} \mathbf{a}t^2$$

Donde:

p': Nueva posición

p: Posición anterior

v: Velocidad actual

t: Tiempo transcurrido entre la última actualización.

a: Aceleración.

Las letras en negrita son vectores.

La actualización de la velocidad se lleva a cabo por la también conocida formula:

$$\mathbf{v}' = \mathbf{v} + \mathbf{a}t$$

Aunque con las ecuaciones anteriores se tiene el comportamiento de un objeto cuando una fuerza está actuando sobre el (la gravedad), no se consideran otras fuerzas, por ejemplo la fuerza que aplica el suelo a la esfera. Se necesitaba un mecanismo que funcionara para varias fuerzas.

La solución fue usar el principio de D'Alembert, el principio simplificado para usarse en nuestra aplicación es:

$$\mathbf{f} = \sum_i \mathbf{f}_i$$

Este principio implica que si tenemos un conjunto de fuerzas actuando en un objeto, las podemos reemplazar por una fuerza única(**f**). Por este cambio se tuvo que agregar un vector que fuera acumulando las fuerzas que se estuviesen ejerciendo en la esfera, con este vector en cada actualización de la esferas se calcula la aceleración correcta.

REACCIONANDO A LAS COLISIONES (TERCERA LEY DE NEWTON)

Una vez que se detecto una colisión, había que reaccionar de forma que el resultado pareciera lo más real posible, esto es, había que calcular que dirección y que velocidad la esfera llevaría después de la colisión, esto se logró con una de las ecuaciones de la tercera ley de newton:

$$v' = -cv$$

Donde v' es la magnitud de la velocidad después de la colisión, v la velocidad antes de la colisión, y c una constante llamada el coeficiente de restitución.

El coeficiente de restitución controla la rapidez de un objeto después de colisionar, depende de los materiales en la colisión, diferentes pares de material tendrán diferentes coeficientes. Algunos objetos como

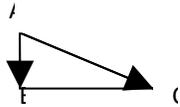
bolas billar, rebotarán. Otros objetos se quedarán pegados cuando colisionen, por ejemplo una bola de nieve lanzada a la cara de alguien.

Si el coeficiente es 1, entonces los objetos rebotarán con la misma rapidez con la que se estaban acercando. Si el coeficiente es 0 los objetos se quedarán pegados.

En este proyecto para lograr el efecto de rigidez se usa un coeficiente igual a 1.

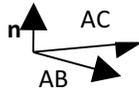
DIRECCIÓN DE LA COLISIÓN

La dirección de la colisión es la dirección en la que el objeto se dirigirá después de la colisión por lo tanto es un vector, este vector se calcula del triangulo detectado en la colisión:



Dicho vector, es el vector normal al triangulo de la colisión, y se calcula con el producto cruz de los vectores **AC** y **AB**:

$$\mathbf{n} = \mathbf{AC} \times \mathbf{AB}$$



Ahora la colisión que la esfera llevará después de la colisión depende de la magnitud de su velocidad, del coeficiente de sustitución y de este vector normal:

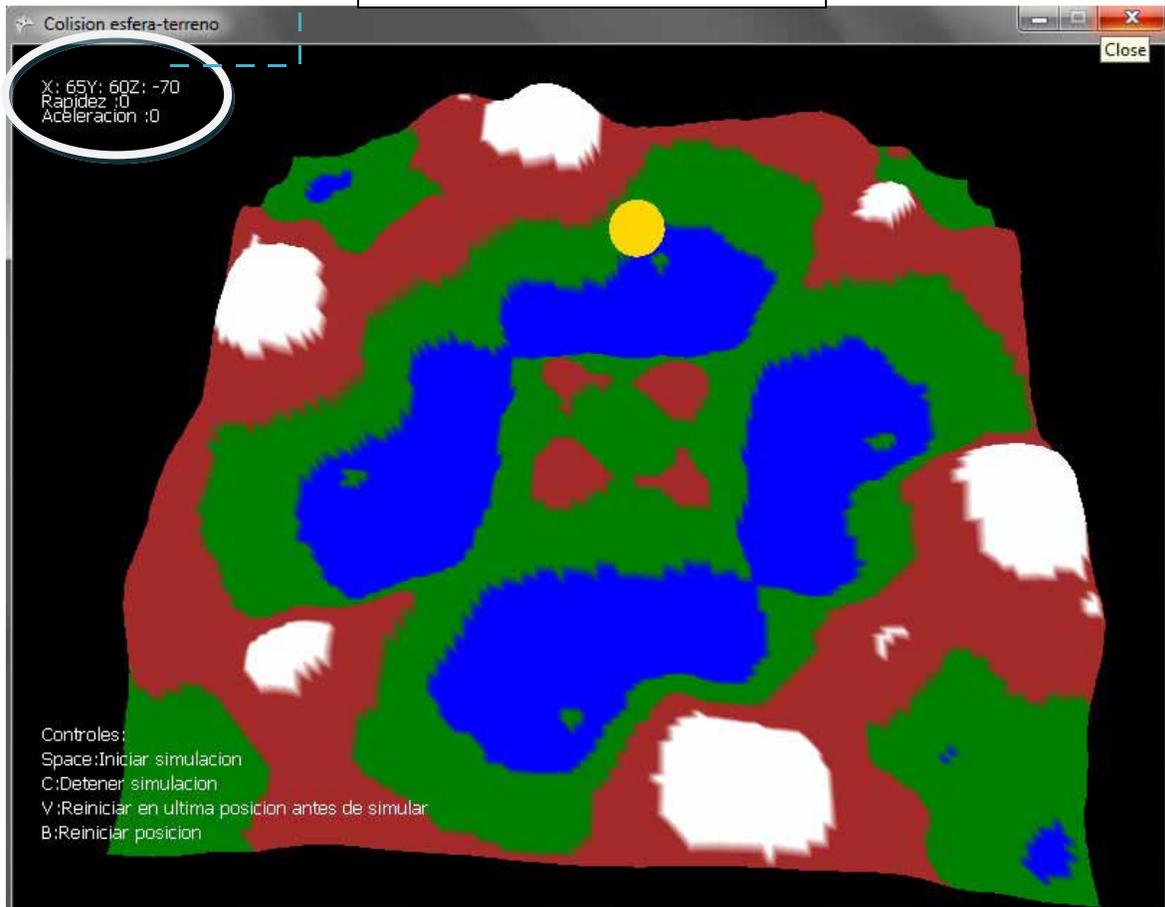
$$\mathbf{v}' = \mathbf{n}|\mathbf{v}|(-c)$$

Donde \mathbf{v}' es el vector después de la colisión, \mathbf{n} el vector normal al triangulo de la colisión, $|\mathbf{v}|$ la magnitud de la velocidad antes de la colisión y c el coeficiente de restitución.

MANUAL DE LA APLICACIÓN

VENTANA DE VISUALIZACIÓN:

Información de la esfera, posición, rapidez y aceleración.

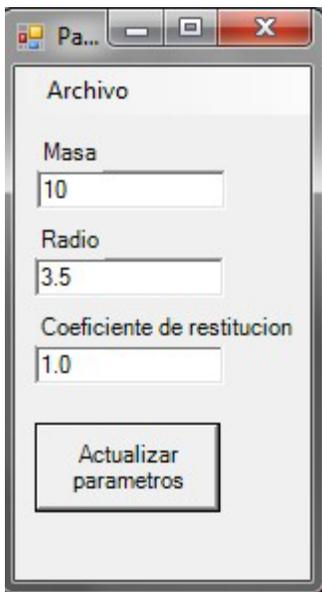


Los controles completos son:

Tecla	Acción
Arriba	Mueve la esfera hacia adelante.
Abajo	Mueve la esfera hacia atrás.
Izquierda	Mueve la esfera hacia la izquierda.
Derecha	Mueve la esfera hacia la derecha.
Re Pag	Mueve la esfera hacia arriba
Av Pag	Mueve la esfera hacia abajo
E	Mueve la cámara hacia adelante.
D	Mueve la cámara hacia atrás.
S	Mueve la cámara hacia la izquierda.
F	Mueve la cámara hacia la derecha.
Q	Mueve la cámara hacia arriba

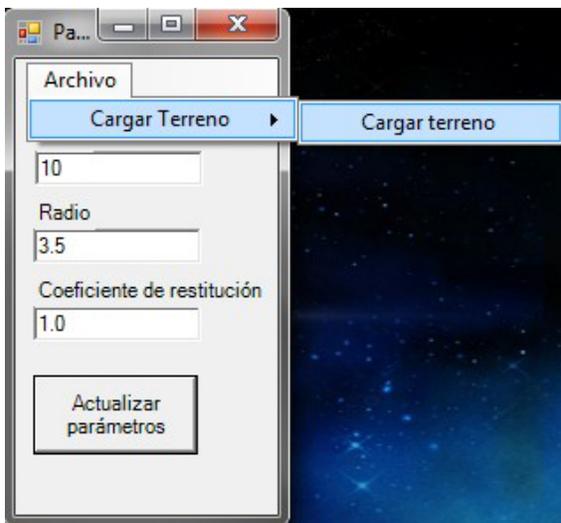
Z	Mueve la cámara hacia abajo.
Barra espaciadora	Comienza la simulación
C	Detiene la simulación
V	Reinicia la posición de la esfera en la última posición antes de comenzar a simular.
B	Reinicia la posición de la esfera

LA VENTANA DE PARÁMETROS DE LA ESFERA



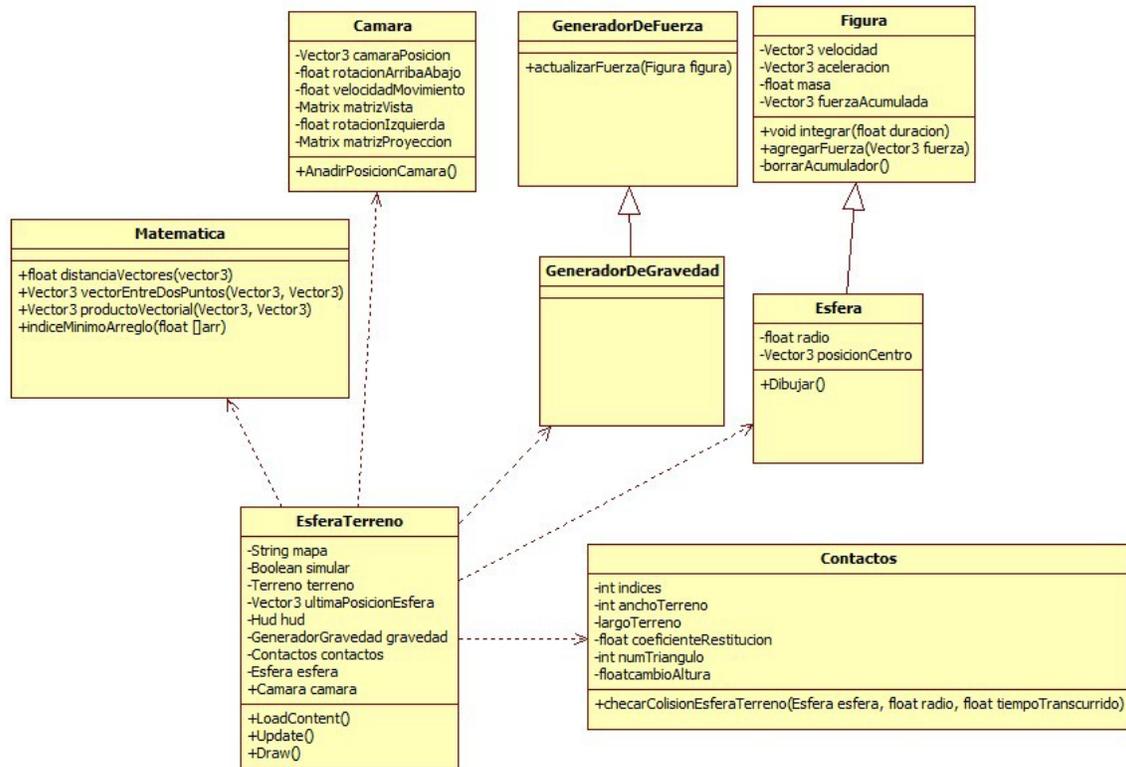
En esta ventana se pueden modificar la masa, radio y coeficiente de restitución de la esfera. Oprimir “actualizar parámetros” para actualizarlos.

CARGANDO OTROS TERRENOS



Para cargar otro terreno ir al menú archivo->Cargar Terreno-> Cargar Terreno. Esto abrirá una ventana para seleccionar un archivo seleccione un con extensión “.bmp”.

DIAGRAMA DE CLASES



DIAGRAMAS DE ENTIDAD-RELACIÓN

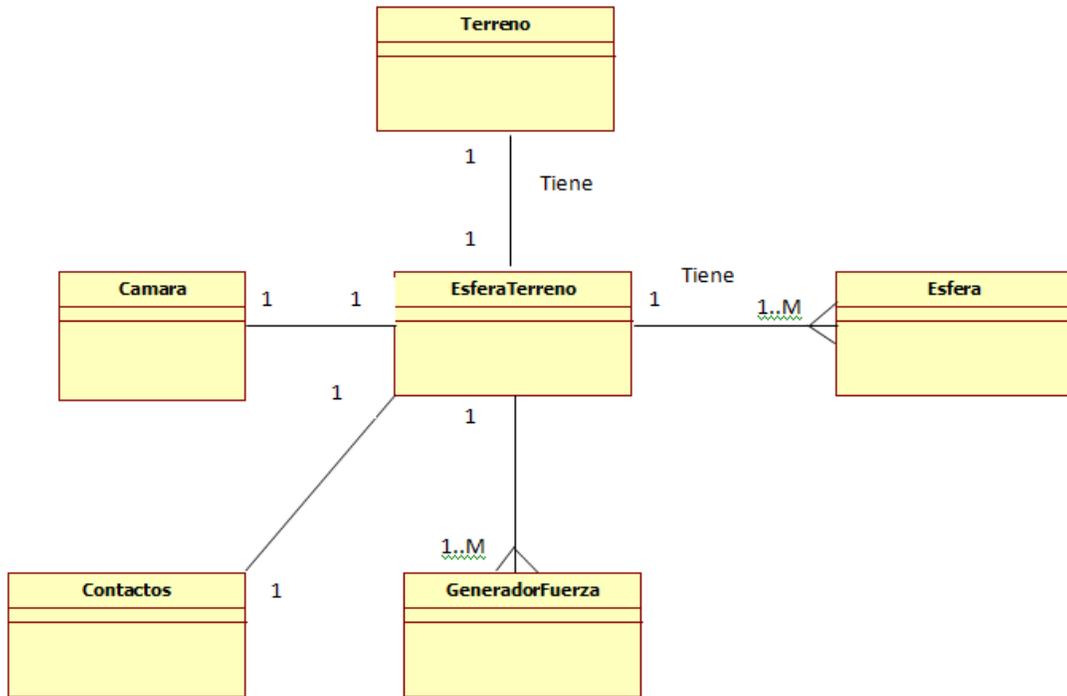


DIAGRAMA ESTADOS

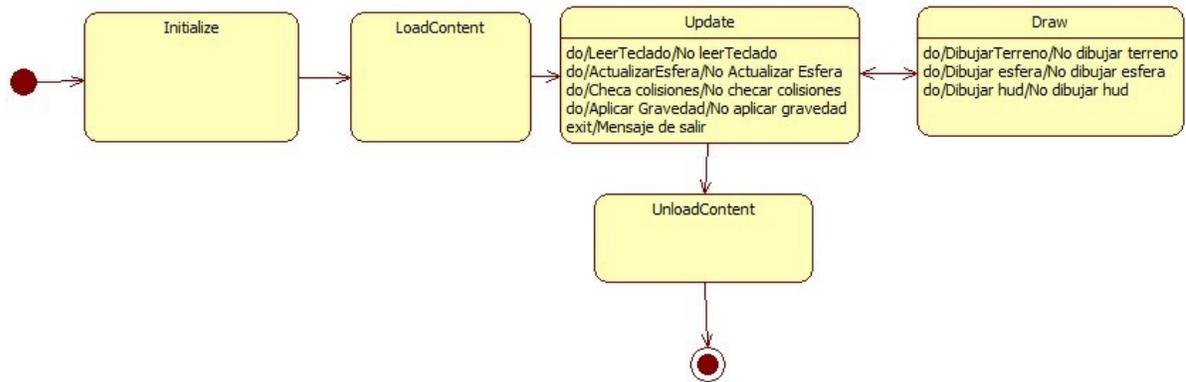


DIAGRAMA DE OBJETOS

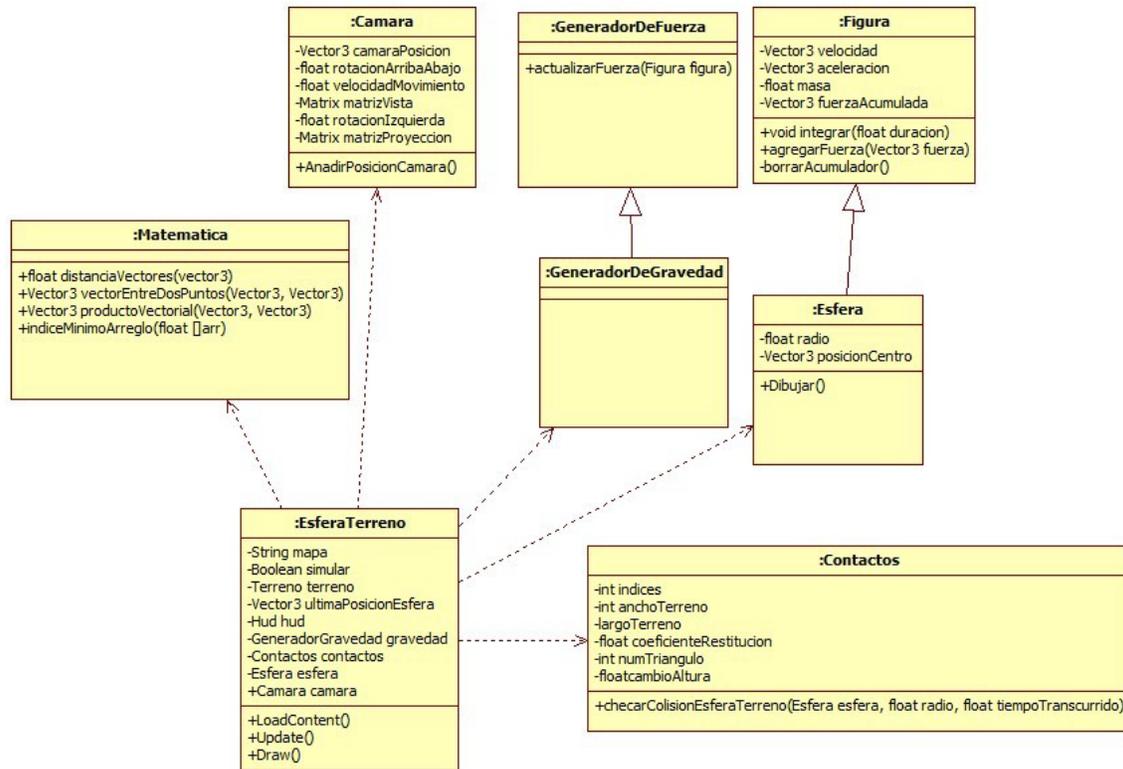
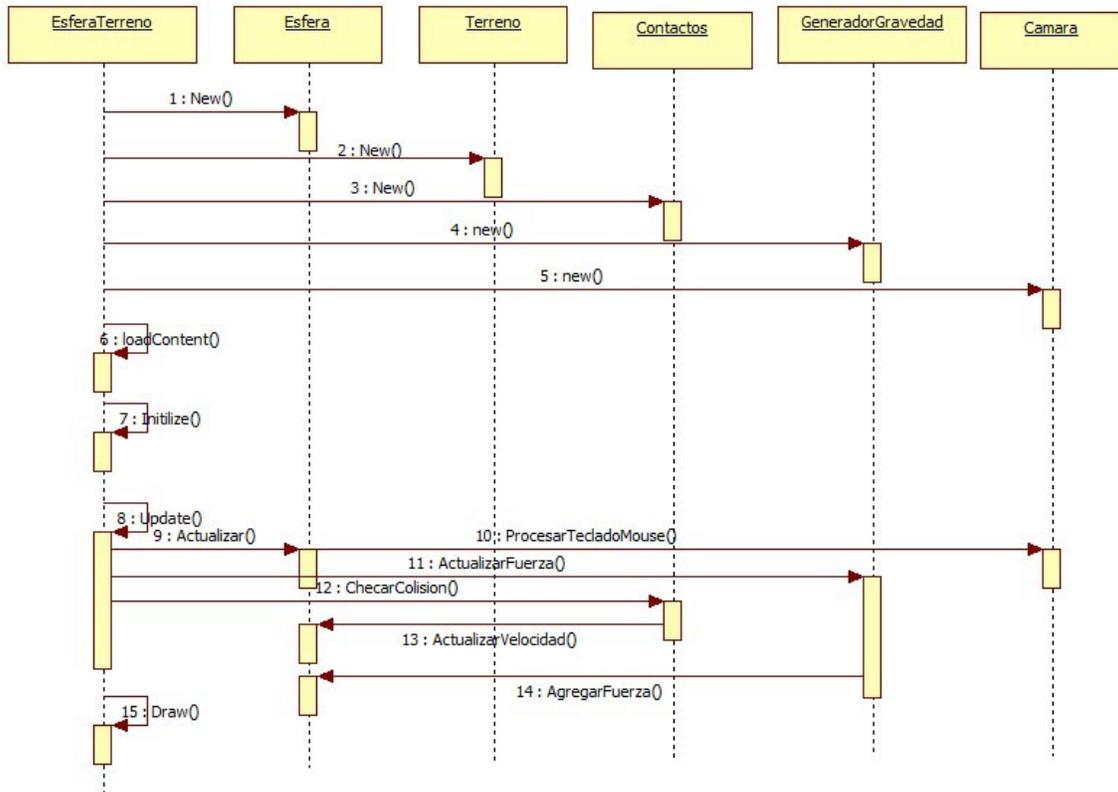


DIAGRAMA DE FLUJO



BIBLIOGRAFÍA:

[1] Paul A. Tipler, Física para la ciencia y la tecnología, Cuarta edición. Volumen 1

[2] <http://www.riemers.net/>

[3] Howard Anton. Introducción al álgebra lineal, sexta edición.