

Documento del Parcial

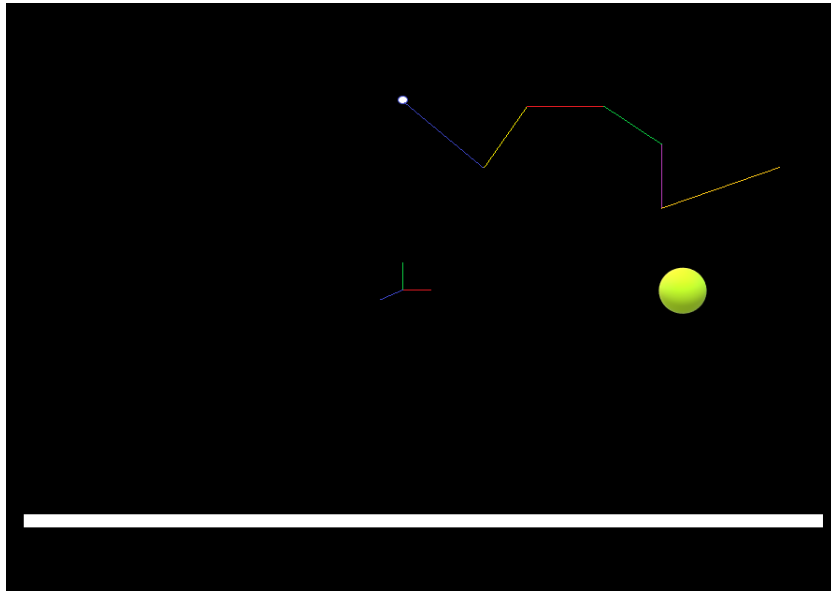
Brazo Robótico

Juan Sebastián Prado Valero

Diseño

Diseño Gráfico:

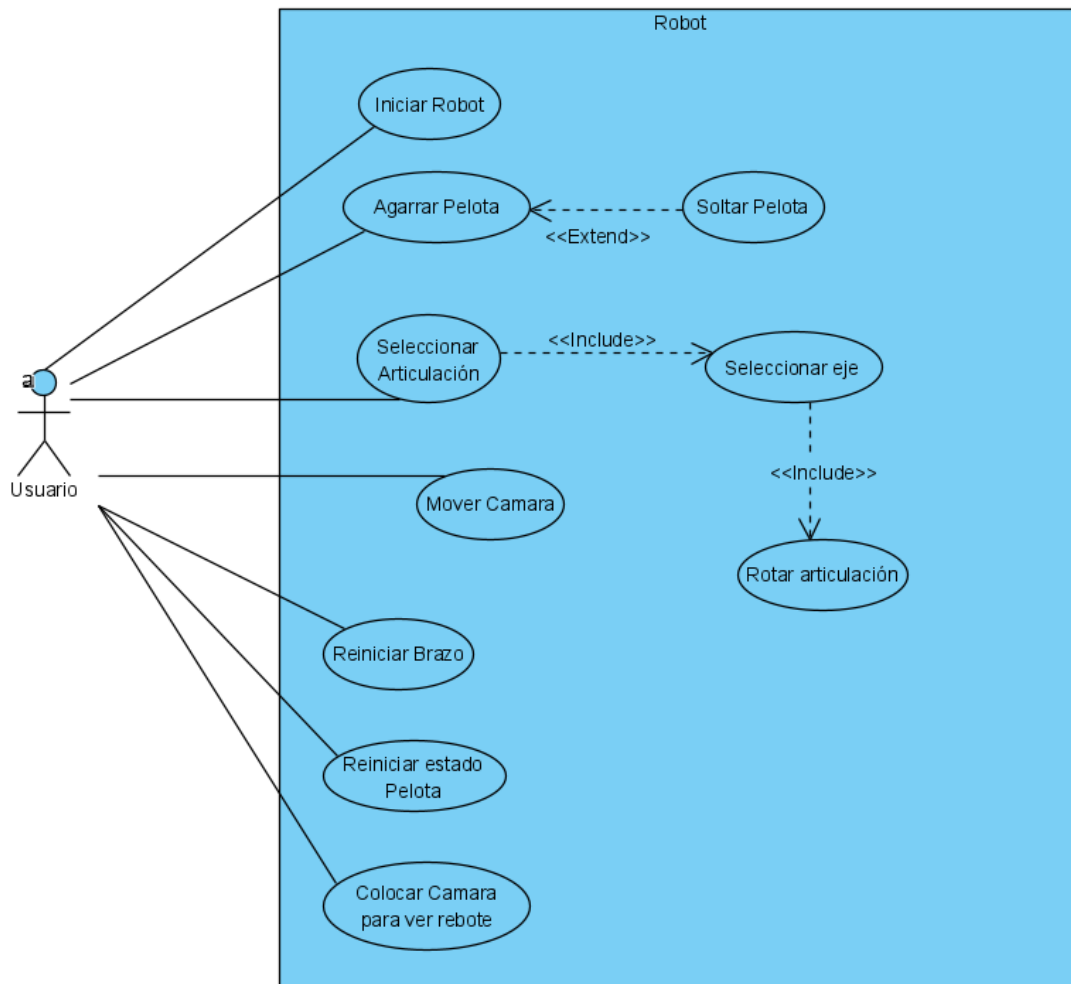
En primera instancia me imagine el resultado del parcial de esta manera:



Donde se representaran las distintas articulaciones con colores para diferenciarlas, además de que la pelota fuera amarilla para poder reconocerla fácilmente, y con un suelo de color blanco representado por un cuadrado en la zona inferior del brazo. Y el posicionamiento pensado fue tener el inicio del brazo en $y = 10$ y que el piso se posicionara de acuerdo con el tamaño del brazo, de igual forma que la pelota se posicionara de manera aleatoria en un radio alrededor del brazo del mismo tamaño de este.

Diseño de Software:

Diagrama de CU



Los casos de uso identificados son:

Iniciar Robot:

En este caso de uso el usuario inicia el programa especificando el archivo que se va a leer desde consola

Mover Cámara:

En este caso de uso el usuario puede mover la posición en la que mira la cámara, con el teclado:

- W para hacer zoom
- S para retroceder el zoom
- A para rotar hacia la izquierda
- D para rotar hacia la derecha

- C para levantar la cámara en el eje y
- V para bajar la cámara en el eje y
- Con el mouse puede mover la cámara dando click izquierdo y arrastrando (dependiendo hacia donde se arrastre el centro del glLookAt se mueve un poco hacia allá)

Seleccionar articulación:

En este caso de uso el usuario puede seleccionar una articulación con el teclado oprimiendo los números del 1 al $1 < n < 10$ donde n es la cantidad de articulaciones ingresadas.

Seleccionar eje:

En este caso de uso el usuario puede seleccionar el eje sobre el cual rotará la articulación seleccionada, si oprime “x”, “y” o “z” rotará en su respectivo eje.

Rotar Articulación:

En este caso de uso el usuario puede rotar la articulación seleccionada en el eje seleccionado +5 grados si oprime “+” o -5 grados si oprime “-”.

Agarrar Pelota:

En este caso de uso el usuario puede agarrar la pelota oprimiendo “p” en caso de que el punto final del brazo robótico se encuentre en contacto con la esfera de la pelota.

Soltar Pelota:

En este caso de uso el usuario puede soltar la pelota en cualquier momento, oprimiendo la letra “p” y esta empezará a caer en caída libre contra el suelo y ahí rebotará varias veces donde perderá un 20% de su velocidad hasta quedar totalmente en el suelo.

Reiniciar Brazo:

En este caso de uso el usuario puede reiniciar la posición de las articulaciones del brazo dando a la tecla “r”.

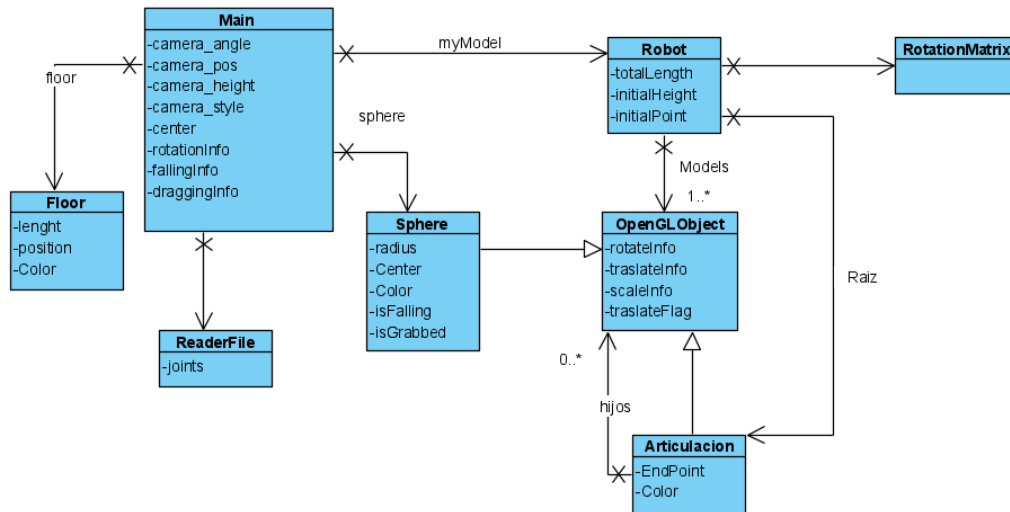
Reiniciar estado pelota:

En este caso de uso el usuario puede reiniciar el cálculo de la posición de la pelota oprimiendo la tecla “b”

Colocar cámara para ver rebote:

En este caso de uso el usuario puede poner una especie de cámara para poder apreciar mejor el rebote de la pelota oprimiendo la tecla “m” y si la vuelve a oprimir el programa trata de reubicar la cámara más cerca al brazo.

Modelo de dominio



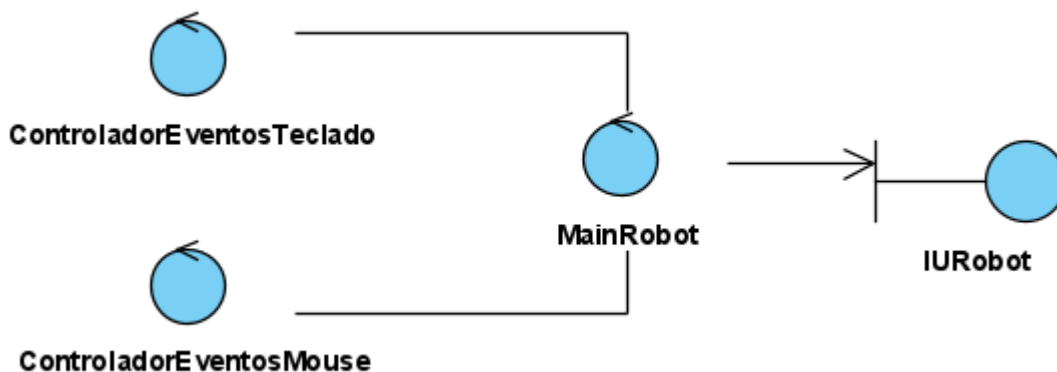
En el modelo de dominio se puede apreciar la forma en la que se maneja la información en la aplicación, donde:

- Main:**
 Contiene información del escenario como el ángulo de la cámara, su posición, su altura y el estilo de esta. También incluye un atributo para conocer el centro al que apunta la cámara, junto a información relacionada con la rotación de alguna articulación, información relacionada con la caída de la pelota, información relacionada con el arrastrado del mouse y entidades que representan el modelo del Robot, de la pelota, el piso y un lector de archivos.
- Robot:**
 El robot contiene información relacionada con su longitud total, el punto inicial de su extremo, y la altura inicial del brazo. Además de manejar una lista de Modelos de tipo OpenGLObject y una raíz que será su articulación inicial, y por ultimo tiene una entidad de RotationMatrix
- RotationMatrix:**
 Se encarga de realizar rotaciones y traslaciones matemáticamente con el fin de ubicar correctamente la pelota y el extremo del brazo robótico
- OpenGLObject:**
 Contiene información relacionada a hacer la rotación, escalamiento y traslación usando las funciones de OpenGL
- Articulación:**
 Es la encargada de armar el árbol de articulaciones y mantener las dependencias entre estas para que al rotar una, roten sus sucesores, sus hijos son de tipo OpenGLObject. Hereda de OpenGLObject.
- Sphere:**
 Es la encargada de dibujar la pelota y realizar cálculos para saber si un punto esta en

contacto con esta, tiene un radio, su centro, su color, un atributo que indica si está cayendo y otra de si está agarrada. Hereda de OpenGLObject.

- **Floor:**
Representa el piso con la longitud de sus lados, su posición en el eje Y y su color
- **ReaderFile:**
Es el encargado de leer el archivo de entrada con la información de las articulaciones

Diagrama de Colaboración (EBC)



La imagen previa representa el flujo de interacción del Robot:

1. El usuario solicita que el programa inicie indicando el archivo de entrada al Main que funciona como un controlador para el manejo de información de los objetos del programa (articulaciones, pelota y el piso)
2. El programa dibuja inicialmente los objetos del programa (IU Robot)
3. El usuario interactúa con su teclado o con su mouse
4. A través de los eventos de teclado y del mouse que maneja OpenGL se obtendrá la información necesaria para realizar los movimientos de cámara, de las articulaciones e interacciones con la pelota
5. El Main al recibir esa información realiza cálculos necesarios para indicar que dibujar
6. El IU Robot es la representación de las funciones de OpenGL que pintan sobre la pantalla

Manual de Usuario

1. Descargar librerías de Python como:
 - a. `math`, `sys`, `re`, `random`, `glm` y `PyOpenGL`
2. Se debe realizar un archivo de extensión `.txt` con la siguiente estructura:

```
n
l_1 l_2 ... l_n
```

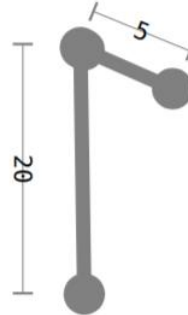
donde:

- $1 < n < 10$ es el número de articulaciones del brazo;
- l_i es la longitud de la i -ésima articulación.

Por ejemplo, el brazo definido por:

```
2
20 5
```

sería:



3. Abrir una consola en la ruta de la carpeta del proyecto
4. Introducir el siguiente comando
 - a. `py Parcial.py <archivo>`

Donde en el último elemento deben colocar un `txt` que contiene la información del robot y que quede de la siguiente manera:

```
Documentos\CG\Parcial>py Parcial.py input.txt_
```

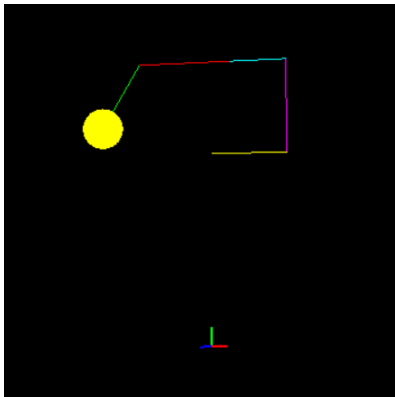
5. Al iniciar el programa se mostrará así:



Donde se dibuja la pelota (de radio del 5% de la longitud del brazo) en una posición aleatoria y se dibujan las articulaciones del brazo del robot (La articulación amarilla es la primera) y debajo del todo se encontrará el piso en forma de Paralelepípedo de

color blanco.

6. Para realizar movimientos de cámara puede usar las siguientes teclas o el mouse:
 - a. W para hacer zoom
 - b. S para retroceder el zoom
 - c. A para rotar hacia la izquierda
 - d. D para rotar hacia la derecha
 - e. C para levantar la cámara en el eje y
 - f. V para bajar la cámara en el eje y
 - g. Con el mouse puede mover la cámara dando click izquierdo y arrastrando (dependiendo hacia donde se arrastre el centro del glLookAt se mueve un poco hacia allá)
7. Para mover una articulación:
 - a. Debe seleccionar una articulación oprimiendo en el teclado los números del 1 al $1 < n < 10$ donde n es la cantidad de articulaciones ingresadas.
 - b. Seleccionar el eje de rotación oprimiendo las teclas X, Y o Z
 - c. Debe oprimir las teclas “+” o “-“ para aumentar o disminuir el ángulo de rotación de la articulación
8. Para interactuar con la pelota:
 - a. Debe posicionar las articulaciones de tal forma que el extremo del brazo robótico esté en contacto con la pelota
 - b. Presione la tecla “p” y verá como el robot agarra la pelota de esta forma



- c. Y puede volver a rotar los brazos para manipular la pelota.
9. Para soltar la pelota:
 - a. Vuelva a presionar la tecla “p” y el brazo soltara la pelota para que esta caiga al suelo
 - b. Puede oprimir la tecla “m” para una mejor visión al rebote de la pelota
10. Para Reiniciar la posición del brazo robótico:
 - a. Presione la tecla “r” y el brazo volverá a como estaba al inicio
11. Para Reiniciar la posición de la pelota de manera aleatoria:
 - a. Presione la tecla “b” y esta se reubicara aleatoriamente dentro de su rango

Conclusión

Durante el desarrollo surgieron varios problemas a los cuales se tuvo que hacer frente con investigación y análisis de transformaciones para el cálculo de las verdaderas posiciones de los objetos y si poder colisionarlos. A pesar de esos problemas me divertí mucho realizando y aprendiendo a la vez que lo hacía, me siento muy alegre de haber podido cumplir este reto que al principio me pareció mucho más complicado y no imaginaba poder hacer esto.

Aprendí sobre colisiones y representación de principios físicos en la computación, también la forma en la que se refrescan los frames por segundo del dibujado de la escena con el rebote y caída de la pelota (me sorprende lo real que se puede lograr ver). También repase la forma en la que se aplican las transformaciones y como afectan el punto en cuestión.

Me divertí bastante en el desarrollo de este brazo robótico.