

# Implementación de un Prototipo

Entorno de Simulación Robótico

Anthony Figueroa  
pgsimrob@fing.edu.uy

## **Tutor**

Gonzalo Tejera

## **Cotutores**

Gustavo Armagno, Facundo Benavides, Serrana Casella

10 de setiembre de 2007

Instituto de Computación  
Facultad de Ingeniería - Universidad de la República  
Montevideo - Uruguay



# Índice

1. Introducción	3
2. Motivación	5
3. Alcance	7
3.1. Configuración del entorno de juego . . . . .	7
3.2. Configuración de los robots . . . . .	7
3.3. Visualizar el estado de la simulación . . . . .	7
3.4. Comunicación asincrónica con procesos . . . . .	7
3.5. Configuración del paso de tiempo . . . . .	7
4. Resultados	9
4.1. Sintaxis de los archivos de configuración . . . . .	9
4.2. Motor físico . . . . .	11
4.3. Motor gráfico . . . . .	12
4.4. Comunicación . . . . .	13
5. Conclusiones	15

## Índice de figuras

1.	Simulación utilizando drawstuff . . . . .	12
2.	Simulación utilizando Irrlicht . . . . .	13

## **1. Introducción**

Este documento tiene como objetivo relatar diversos aspectos de la creación de un prototipo implementado en el marco de este proyecto de grado. En la sección 2 se describen los diversos motivos que justificaron la creación de un prototipo. En la sección 3 se explicará con detalle el alcance de esta propuesta y los objetivos buscados. En la sección 4 se realizará una descripción de los resultados obtenidos, los cuales incluyen un relato de las experiencias adquiridas y los problemas que se encontraron. Finalmente, en la sección 5 se describirán las conclusiones alcanzadas en el contexto de este prototipo.



## 2. Motivación

La principal motivación para la creación de este prototipo es la necesidad de reducir riesgos técnicos en diversas áreas. Por un lado, se deseaban tener elementos para lograr realizar una correcta elección de las librerías a utilizarse en este proyecto. Estas librerías se utilizarán para implementar el motor físico y el motor gráfico del sistema final, por lo que es de vital importancia elegir las mismas de forma correcta. Además se buscaba obtener experiencia en el manejo de dichas librerías, especialmente en el manejo del motor físico del simulador, el cual presenta cierto grado de complejidad. Se deseaba elegir el motor físico de tal manera que con el mismo se logre cumplir con todos los requerimientos del sistema. Por otro lado se deseaba estudiar la forma como se realiza la integración entre el componente físico y el componente gráfico del simulador, con el objetivo de independizar ambos componentes.

Finalmente, otro aspecto que motivó fuertemente la creación de este prototipo, fue el deseo de crear un sistema simplificado que cumpliera con buena parte de los requerimientos más críticos del sistema final.



### **3. Alcance**

En este prototipo se planteó la creación de una simulación de robots simplificada que presentara las características que se describirán a continuación.

#### **3.1. Configuración del entorno de juego**

El modelo físico del entorno de juego debe ser configurable. El mismo debe ser obtenido por el simulador a través de un archivo hecho por el usuario. Para esto se definió una sintaxis en la cual se define el terreno de juego. Como una simplificación, el entorno de juego debe estar compuesto simplemente por un conjunto de cajas, a las cuales se le puede especificar diversos parámetros.

#### **3.2. Configuración de los robots**

El modelo físico de los robots debe ser configurable. El mismo al igual que el entorno de juego debe ser obtenido desde un archivo. Para esto se realizaron diversas simplificaciones. En primer lugar, el chasis del robot está compuesto por una caja a la cual se le especifican sus características. Además, solo se requiere un tipo de empalme, el cual conecta al chasis con una rueda. El robot puede tener cualquier cantidad de ruedas, a las cuales se les puede especificar sus características, cada una de dichas ruedas con su respectivo empalme conectándolo con el chasis. A este empalme también se le puede especificar sus características, como sus ejes.

#### **3.3. Visualizar el estado de la simulación**

El estado de la simulación debe poder ser visualizado a través de una interfaz gráfica. La misma debe ser una librería externa al motor físico, por lo que se debe estudiar su integración. No es deseable que exista una dependencia entre esta librería y el motor físico elegido para esta implementación en particular.

#### **3.4. Comunicación asincrónica con procesos**

Para controlar los robots se debe implementar un protocolo de comunicación simple con un proceso que los controle. El mismo debe obtener el estado del mundo comunicándose con el simulador y enviarle información acerca de la velocidad que se desea aplicar a cada rueda.

#### **3.5. Configuración del paso de tiempo**

El simulador debe proveer la funcionalidad de configurar el paso del tiempo. Esto significa que se debe poder avanzar la simulación a velocidad avanzada o también hacerlo paso a paso. Se deben estudiar los límites que deberán existir sobre este requerimiento y los efectos no deseados que pueden llegar a ocurrir al no respetar dichos límites.



## 4. Resultados

### 4.1. Sintaxis de los archivos de configuración

En este prototipo en particular se creó un archivo que define una cancha análoga a la utilizada por el simulador de la FIRA[RSOCF] en la categoría MiroSot Middle League [FIRA].

La descripción de las características del entorno de juego esta delimitada por los tags <Field>y </Field>. La descripción de una caja se realiza entre los tags <Box>y </Box>. Entre los tags que delimitan el entorno de juego pueden haber tantas descripciones de cajas como se deseen. La descripción de de dichas cajas comprende parámetros para definir las dimensiones, posición y orientación. Cabe destacar que las cajas que modelan el entorno son estáticas, por lo que no se le debe especificar una masa.

La sintaxis definida, con la cual se define el entorno fue la siguiente:

```
<Field>
<Box>...</Box>
<Box>...</Box>
<Box>...</Box>
.
.
.
</Field>
```

A cada caja se le debe especificar sus características siguiendo la siguiente sintaxis:

```
<Box>
<size_x>x</size_x>
<size_y>y</size_y>
<size_z>z</size_z>
<pos_x>px</pos_x>
<pos_y>py</pos_y>
<pos_z>pz</pos_z>
<roll>r</roll>
<yaw>y</yaw>
</Box>
```

La descripción del conjunto de robots de la simulación sigue la siguiente sintaxis general:

```
<Team>
<Robot>..... </Robot>
<Robot>..... </Robot>
.
.
.
</Team>
```

Se puede especificar tantos robots como se deseen, y la información de cada robot se especifica entre los tags `<Robot>` y `</Robot>`. Esta información debe seguir la siguiente sintaxis:

```

<Id>id</Id>
<Chassis>
<Box>... </Box>
</Chassis>
<Wheel>.... </Wheel>
<Wheel>.... </Wheel>
.
.
.
<WheelJoint>... </WheelJoint>
<WheelJoint>... </WheelJoint>
.
.
.

```

De esta manera se especifican diversas características del robot. Cabe destacar que como una simplificación, el chasis solo puede estar compuesto por una caja, y ningún otro tipo de objeto, a la cual se le puede especificar diversos parámetros como su masa, dimensiones, posición y orientación.

La información de las ruedas se especifica entre los tags `<Wheel>` y `</Wheel>`. Se debe especificar diversos parámetros, como un identificador de la rueda, el radio, la masa, la posición y rotación. Su sintaxis es la siguiente:

```

<Wheel>
<id>idw</id>
<mass>m</mass>
<radius>r</radius>
<width>w</width>
<pos_x>px</pos_x>
<pos_y>py</pos_y>
<pos_z>pz</pos_z>
<roll>rl</roll>
<yaw>y</yaw>
</Wheel>

```

Por otra lado la descripción del empalme que une a cada rueda con el chasis debe ser especificado entre los tags `<WheelJoint>` y `</WheelJoint>`. Se debe especificar el identificador de la rueda con la cual está conectada, así como también una serie de parámetros de configuración. Se debe especificar ambos ejes de este empalme. El eje 1 corresponde al sentido perpendicular al eje de giro de la rueda, normalmente el eje perpendicular al piso. El eje 2 corresponde al eje sobre el cual gira la rueda. Su sintaxis es la siguiente:

```

<WheelJoint>

```

```

<axis1_x>a1x</axis1_x>
<axis1_y>a1y</axis1_y>
<axis1_z>a1z</axis1_z>
<axis2_x>a2x</axis2_x>
<axis2_y>a2y</axis2_y>
<axis2_z>a2z</axis2_z>
<bounciness>b</bounciness>
<hardness>h</hardness>
<fmax>f</fmax>
<wheelid>idw</wheelid>
</WheelJoint>

```

## 4.2. Motor físico

Entre los diversos motores físicos que podrían llegar a cumplir los requerimientos de este proyecto, en primer lugar se decidió utilizar la librería Newton Game Dynamics[NGD07]. Se tomó esta decisión con base en las buenas características de esta librería ya estudiadas en [AF07]. Fundamentalmente, la flexibilidad en el manejo del paso del tiempo y la facilidad de uso la transformaba en una interesante alternativa.

En un principio se notó un buen comportamiento del modelo físico del simulador llevado a cabo utilizando esta librería. Sin embargo al realizar pruebas utilizando robots pequeños, análogos a los robots de la categoría MiroSot Middle League de la FIRA[FIRA], se observó que esta librería presentaba serios problemas de estabilidad numérica. Es probable que ajustar los parámetros de la simulación de tal manera que se logre una correcta simulación sea posible, pero la escasa experiencia en la utilización de esta librería no permite asegurar con certeza dicha hipótesis. De esta manera por lo tanto, se descartó la utilización de dicho motor físico.

La siguiente alternativa que se evaluó fue la utilización de la librería ODE. Esta librería no presenta tanta flexibilidad en el manejo del paso del tiempo como la anterior alternativa, pero presenta diversas cualidades que implican que esta sea una alternativa a tener en cuenta. Se logró modelar una correcta simulación utilizando esta librería. Se logró que utilizando los archivos de configuración, se creara un modelo físico correspondiente a la configuración expresada a través de los mismos. Por otro lado, no se observaron inestabilidades numéricas serias como se observaron con la anterior alternativa. Se probaron diversas configuraciones de robots y diversas cantidades de los mismos y se obtuvieron resultados satisfactorios. Además, se logró cumplir los requerimientos de configuración del avance del paso del tiempo, logrando avanzar el mismo de paso a paso, o en velocidad avanzada. Se logró avanzar la simulación a velocidades cercanas al triple del paso normal sin mayores problemas.

Una consideración muy importante a tener en cuenta es el hecho de que ODE debe utilizar lapsos de tiempo de largo fijo para avanzar la simulación. Esto significa que no se puede cambiar este período en cada paso de la iteración,

sino que debe ser siempre el mismo. Utilizar largos variables, lleva según su autor, a efectos no deseados en las simulaciones. Sin embargo, un requerimiento importante del sistema es la manipulación del paso del tiempo, pero el cambio del período de avance de la simulación ocurre en un solo paso de la misma, por lo que no se observan efectos no deseables que perduren en las siguientes iteraciones.

Un punto a tener en cuenta es que la velocidad con la que se ejecuta la simulación puede variar notablemente dependiendo de las prestaciones de la computadora sobre la cual se ejecuta. Por lo tanto, es deseable que el sistema final posea una instancia de configuración para adaptar el valor del período de tiempo de avance de la simulación a la velocidad de ejecución deseada.

El requerimiento de manipulación del paso de tiempo, fue cumplido satisfactoriamente utilizando esta librería.

### 4.3. Motor gráfico

La librería física utilizada incluye una librería gráfica muy fácil de utilizar denominada "drawstuff", pero acoplada fuertemente al componente físico específico. Fue muy útil su utilización en las pruebas realizadas sobre el motor gráfico, sin embargo un interesante desafío era integrar la librería ODE con una librería gráfica externa, manteniendo la independencia entre estos módulos. Se utilizó la librería Irrlicht[IRRL], la cual resultó de fácil utilización y integración a la simulación física, ya que presenta diversos ejemplos y buena documentación en su sitio oficial. Especialmente presenta documentos de como integrarla con diversas librerías físicas, incluyendo ODE.

Es importante destacar que antes de iniciar la simulación es posible elegir el método de visualización deseado, que puede ser drawstuff o Irrlicht. A continuación se puede ver imágenes de ambas alternativas en una simulación utilizando 3 robots.



Figura 1: Simulación utilizando drawstuff

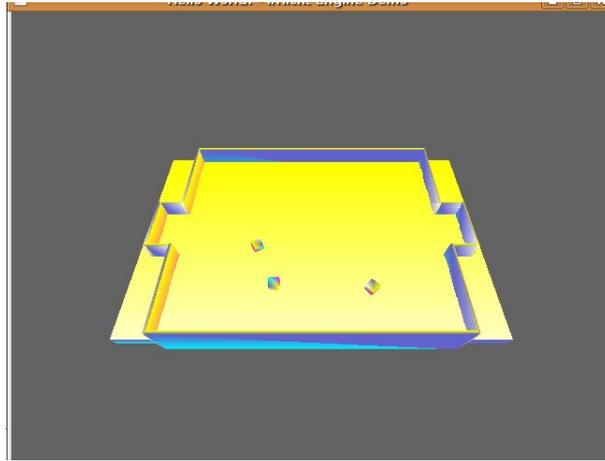


Figura 2: Simulación utilizando Irrlicht

#### 4.4. Comunicación

Se implementó una estrategia muy simple para controlar los robots. Su funcionamiento consiste simplemente en enviar velocidades aleatorias dentro de un rango dado. Para realizar la comunicación de dicha estrategia con el simulador, en primer lugar se implementó un simple algoritmo sincrónico, en el cual el simulador iniciaba la ejecución de un hilo, y en este hilo enviaba la información del estado de los robots y aguardaba la respuesta de forma sincrónica. Esta respuesta contenía las velocidades de las ruedas de los robots generado en el proceso que implementaba una estrategia simple. Se obtuvieron resultados satisfactorios en este prototipo de comunicación.

Luego de esto, se implementó un nuevo protocolo de comunicación, que tiene como característica principal el hecho que sea asincrónico. El simulador crea dos hilos nuevos que se utilizan para la comunicación con la estrategia. En uno de los hilos se hace difusión del estado del mundo a través de multicast. En el otro hilo se reciben paquetes, que en este caso son los enviados por el proceso que controla los robots. Dicho proceso inicia un hilo que recibe información del mundo a través de multicast y actualiza su información acerca del estado del mundo. Por otra parte, se envía a cada cierto lapso de tiempo un paquete que contiene las velocidades deseadas de los robots, que es recibido por el simulador.

Se obtuvieron resultados satisfactorios, ya que la comunicación funcionó correctamente y no se detectaron desviaciones del comportamiento normal del simulador por el hecho de utilizar recursos computacionales en tareas de comunicación.



## 5. Conclusiones

Se cumplieron satisfactoriamente con los requerimientos propuestos en el contexto de este prototipo. Se obtuvo experiencia en el manejo de librerías claves, lo que será de gran importancia en el desarrollo del sistema final. Por otra parte, se considera que los requerimientos propuestos para el desarrollo de este prototipo fueron acertados, ya que contribuyeron a la mitigación de diversos riesgos técnicos y implicaron investigación en puntos claves. Además, se generó código que probablemente sea reutilizado, principalmente en lo que respecta a la utilización del motor físico elegido.



## Referencias

- [Smi07] Rusell Smith, Open dynamics engine v0.8 user guide, Mayo 2007.
- [AF07] Anthony Figueroa, Estado del Arte, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Uruguay, 2007.
- [NGD07] Sitio Web Newton Dynamics Engine, <http://www.newtongamedynamics.com>, Mayo 2007.
- [NWI07] Sitio Web Newton Wiki, [http://walaber.com/newton\\_wiki/index.php](http://walaber.com/newton_wiki/index.php), Mayo 2007.
- [NHC07] Sitio Web <http://www.runehunter.phpnet.us/NewtonHelpCode.html>, Mayo 2007.
- [FIRA] Sitio Web FIRA. <http://www.fira.net>, Junio 2007 .
- [RSOCF] Robot Soccer v1.5a [http://www.fira.net/soccer/simurosot/R\\_Soccer\\_v15a\\_030204.exe](http://www.fira.net/soccer/simurosot/R_Soccer_v15a_030204.exe), Mayo 2007.
- [IRRL] Sitio web <http://irrlicht.sourceforge.net>, Mayo 2007.