Verificación

Entorno de Simulación Robótico

Anthony Figueroa

pgsimrob@fing.edu.uy

Tutor

Gonzalo Tejera

Cotutores

Gustavo Armagno, Facundo Benavides, Serrana Casella

15 de febrero de 2008

Instituto de Computación

Facultad de Ingeniería - Universidad de la República

Montevideo - Uruguay

Índice

1. Introducción

Este documento tiene como objetivo detallar los procedimientos utilizados para comprobar la correctitud del sistema implementado y detectar posibles fallas. En la siguiente sección se detalla el mecanismo utilizado para llevar a cabo estas pruebas, las distintas funcionalidades que se desean verificar y las razones por las cuales las pruebas realizadas son significativas para realizar la verificación del sistema. En las demás secciones se describen cada una de las pruebas realizadas sobre el sistema y el resultado obtenido.

2. Mecanismo de verificación

Una vez que el sistema se encuentra en ejecución, el funcionamiento del mismo puede describirse como una serie de funcionalidades o comandos que son ejecutados por el núcleo de simulación. Es importante que la verificación realizada cubra todas estas funcionalidades y además garantice que el comportamiento de los objetos dentro de la simulación sea razonable. Idear un mecanismo mediante el cual se pudiera garantizar la fidelidad física de la simulación seria una tarea demasiado compleja, por lo cual solamente se espera que los objetos se comporten de una manera razonable. Verificar que las diferentes funcionalidades y combinaciones de funcionalidades funcionen de manera correcta seria de vital importancia para garantizar la estabilidad del sistema.

El punto de partida para la ejecución de comandos es un mensaje recibido por el sistema a través del protocolo UDP. Este mensaje es decodificado y el comando deseado es identificado y ejecutado. Cada vez que en este documento se indique que un agente remoto ejecuta un comando, en realidad se quiere decir que este agente envía un mensaje que desencadena la ejecución de un comando una vez decodificado dicho mensaje por el sistema.

Por otra parte se debe garantizar el correcto funcionamiento del sistema en situación de gran carga de trabajo. Esto significa que debe ser posible simular realidades con gran cantidad de objetos que interactúen entre si, y de alguna medir la manera como se deteriora la performance del sistema en función de la cantidad de objetos presentes en la simulación.

3. Pruebas Funcionales

Las pruebas funcionales del sistema consiste en la construcción de diversas simulaciones, y comprobar el funcionamiento de cada una de las distintas funcionalidades, así como también las diferentes combinaciones de funcionalidades que aporten información acerca del correcto funcionamiento del sistema, o puedan dar indicios de posibles errores.

Para realizar esta pruebas lo mas importante que se debe tener en cuenta son las distintas funcionalidades o comandos que son soportados por el sistema. Existen comandos de diferentes niveles de complejidad y existen ciertas dependencias entre comandos, por ejemplo, para que cierto comando tenga sentido y produzca algún efecto, otro comando debió ser ejecutado anteriormente.

Los comandos soportados por el sistema se listan a continuación, una explicación mas detallada de los mismos se puede encontrar en la documentación final del sistema.

- Registro de agente
- Información de agente
- Creación de entidad: Este comando es el mas complejo del sistema. Actualmente soporta tres diferentes alternativas de creación de entidades:
  - Tipo Robot, que consta de un objeto complejo y genérico, cuyos detalles no corresponden a este documento.
  - Tipo pelota, que consta de una esfera simple.
  - Tipo campo de juego, que consta de una serie de cuerpos simples, sin ningún tipo de empalmes.
- Registro de materiales e interacciones
- Inicializar simulación
- Especificar posición de entidad
- Controlar motores de entidad
- Especificar modo de paso de tiempo
- Especificar bandera de paso de tiempo
- Especificar modo de comunicación
- Especificar tiempo de espera en comunicación sincrónica
- Especificar tamaño del paso de simulación
- Especificar gravedad
- Especificar restricciones de dureza global

- Especificar restricciones de esponjosidad global
- Sincronización con iteración
- Guardar simulación a archivo
- Cargar simulación desde archivo
- Especificación de modo rápido
- Especificación de pasos de tiempo por iteración

3.1. Casos de prueba

El punto mas importante en el momento de especificar casos de prueba es encontrar conjuntos de combinaciones de ejecución de comandos que logren cubrir todos los ciclos de ejecución interesantes y que tengan buenas probabilidades de encontrar fallas en el sistema. La conexión entre agentes y el sistema se realiza de manera asincrónica, salvo en los casos de prueba que se lo indique explícitamente.

3.1.1. Caso 1

En el primer caso de prueba realizado se pretende comprobar funcionalidades básicas del sistema. En primer lugar se inicia el sistema y se procede a registrar dos agentes en modo asincrónico. Luego uno de ellos realiza una petición de información acerca del otro agente conectado.

Por lo tanto el orden de la secuencia de comandos que se ejecutan es la siguiente:

- Registro de Agente (2 veces)
- Información de Agente.

Para comprobar la correctitud de la respuesta obtenida, se corroboran los datos recibidos por el agente que solicita información con los datos reales del agente del cual se solicitan los datos.

3.1.2. Caso 2

En este caso de prueba se inicia el sistema y se procede a registrar dos agentes, uno en modo sincrónico y el otro en modo asincrónico. Aparte de esto se procede de forma análoga al caso anterior.

Por lo tanto el orden de la secuencia de comandos que se ejecutan es la siguiente:

- Registro de Agente (2 veces)
- Información de Agente.

3.1.3. Caso 3

En este caso de prueba se inicia el sistema y se procede a registrar dos agentes en modo sincrónico. Aparte de esto se procede de forma análoga al caso anterior.

Por lo tanto el orden de la secuencia de comandos que se ejecutan es la siguiente:

- Registro de Agente (2 veces)
- Información de Agente.

3.1.4. Caso 4

Este caso de prueba comienza con la ejecución del caso de prueba anterior. Luego uno de los agentes ejecuta el comando de creación de una entidad de tipo robot, una entidad de tipo pelota y una entidad de tipo campo de juego. El robot consta de un cubo con 2 ruedas, y un motor en cada una de las ruedas. El campo de juego consta de prismas con masa infinita. Finalmente se ejecuta el comando de inicialización de la simulación. Por lo tanto el orden de la secuencia de comandos que se ejecutan es la siguiente:

- Caso de prueba 1
- Creación de entidad, tipo robot
- Creación de entidad, tipo campo de juego
- Creación de entidad, tipo pelota
- Inicializar simulación

Para comprobar empíricamente la correctitud de este caso de prueba se utiliza la interfaz gráfica que provee el simulador en la cual se puede observar los distintos objetos de simulación. El resultado esperado es la creación de los objetos mencionados y una correcta interacción entre ellos, incluyendo el impacto del robot y la pelota con el campo de juego debido a la acción de la fuerza gravitatoria. Adicionalmente se debe observar que los agentes conectados al simulador reciban correctamente información acerca del estado del mundo.

3.1.5. Caso 5

Este caso de prueba comienza con la ejecución del caso de prueba 1. Luego alguno de los agentes ejecuta el comando de registro de materiales y interacciones. Para este caso se registra 2 materiales y las interaccione entre ellos. Luego uno de los agentes ejecuta el comando de creación de una entidad de tipo robot, una entidad de tipo pelota y una entidad de tipo campo de juego. Todas las entidades son análogas al caso de prueba anterior pero con la diferencia que se especifica los materiales de los mismos. Luego se ejecuta el comando de inicialización de la simulación y finalmente uno de los agentes ejecuta el comando para controlarlos motores del robot recién creado.

Por lo tanto el orden de la secuencia de comandos que se ejecutan es la siguiente:

- Caso de prueba 1
- Registro de materiales e interacciones
- Creación de entidad, tipo robot
- Creación de entidad, tipo campo de juego
- Creación de entidad, tipo pelota
- Inicializar simulación
- Controlar motores de entidad

Para comprobar empíricamente la correctitud de este caso de prueba en parte se procede en forma análoga al caso de prueba anterior, observando los resultados en la interfaz de usuario. Este caso de prueba se ejecuta diversas veces con distintos valores para los distintos parámetros que regulan las colisiones y se observa que las interacciones entre materiales cambie de forma acorde a lo que se espera para las variaciones efectuadas entre distintas ejecuciones de casos de prueba. Adicionalmente se utiliza una técnica de caja blanca para comprobar el correcto funcionamiento de este caso de prueba. Cada vez que ocurre una colisión entre dos objetos se registra en un archivo los parámetros elegidos para resolver la colisión. Esto se logra agregando el código necesario para registrar estos datos al código del manejador de colisiones. Además, en este caso de prueba ambos robots se deberían comenzar a mover de manera acorde a las velocidades especificadas por los agentes al ejecutar el comando.

3.1.6. Caso 6

Este caso de prueba es similar al caso de prueba anterior, con la diferencia que un agente esta conectado en modo sincrónico y el otro en modo asincrónico.

Por lo tanto el orden de la secuencia de comandos que se ejecutan es la siguiente:

- Caso de prueba 2
- Registro de materiales e interacciones
- Creación de entidad, tipo robot
- Creación de entidad, tipo campo de juego
- Creación de entidad, tipo pelota
- Inicializar simulación
- Controlar motores de entidad

Sin embargo como el modo de ejecución por defecto del simulador es asincrónico, y este modo no es modificado en este caso, no deberían haber diferencias en el comportamiento del sistema entre este caso de prueba y el anterior.

3.1.7. Caso 7

Este caso de prueba es similar al caso de prueba anterior, con la diferencia que ambos agentes están conectados en modo sincrónico.

Por lo tanto el orden de la secuencia de comandos que se ejecutan es la siguiente:

- Caso de prueba 3
- Registro de materiales e interacciones
- Creación de entidad, tipo robot
- Creación de entidad, tipo campo de juego
- Creación de entidad, tipo pelota
- Inicializar simulación
- Controlar motores de entidad

Sin embargo como el modo de ejecución por defecto del simulador es asincrónico, y este modo no es modificado en este caso, no deberían haber diferencias en el comportamiento del sistema entre este caso de prueba y el anterior.

3.1.8. Caso 8

Este caso de prueba comienza con la ejecución del caso de prueba 5. Luego uno de los agentes ejecuta el comando de creación de entidad, para el tipo de entidad robot, registrando de esta manera un nuevo robot en la simulación luego de inicializada la misma. Luego se desea manipular el modo de paso de tiempo para comprobar su correcto funcionamiento. Al comienzo el modo de paso de tiempo es el modo normal. En ese momento ejecutar el comando para cambiar la bandera de paso de tiempo no tiene ningún efecto sobre la simulación. Después de esto se ejecuta el comando de cambiar el modo de paso de tiempo para el modo paso a paso. A partir de este momento la simulación se encontrará en pausa. Para avanzar un paso en la simulación se debe ejecutar el comando para especificar la bandera de paso de tiempo. Este comando se ejecuta diversas veces para comprobar que a cada ejecución del comando la simulación avanza un paso. Luego de esto se modifica el modo de paso de simulación nuevamente, llevándolo al modo normal. A continuación se pasará el simulador a modo rápido, en el cual los cálculos necesarios para avanzar un paso de simulación se hacen de manera tal que se pierde precisión y se gana velocidad. Finalmente se especifica la cantidad de pasos por iteración, aumentándolo desde 1 (por defecto) hasta algún valor determinado.

- Caso de prueba 5
- Especificar modo de paso de tiempo (paso a paso)
- Especificar bandera de paso de tiempo (varias veces)
- Especificar modo de paso de tiempo (normal)
- Especificación de modo rápido
- Especificación de pasos de tiempo por iteración

En este caso de prueba se puede observar a través de la interfaz de usuario si el comportamiento del simulador es el deseado. Cuando se especifica el modo paso a paso la simulación debe permanecer en pausa. Cuando en este modo se especifica la bandera de paso de tiempo, el simulador debe avanzar un iteración la simulación. Una vez que se vuelve al modo normal, la simulación debe seguir su funcionamiento de manera convencional. Para comprobar que la ejecución se hace en modo rápido nuevamente se utiliza una técnica de caja blanca, agregando banderas que se modifican al momento de realizar el paso de simulación y se asignan valores diferentes en ambos modos, y controlando el valor de dicha bandera se puede ver que el sistema realmente se ejecuta en modo rápido. Además, al aumentar la cantidad de pasos por iteración el incremento en la velocidad de la simulación debe ser proporcional al aumento efectuado.

3.1.9. Caso 9

Este caso de prueba comienza con la ejecución del caso de prueba 5. Luego uno de los agentes ejecuta el comando de creación de entidad, para el tipo de entidad robot, registrando de esta manera un nuevo robot en la simulación luego de inicializada la misma. Después de esto luego de unos instantes, uno de los agentes modifica la posición de un robot y de la pelota, ubicándolos en una nueva posición dentro de la simulación. El otro agente conectado le envía comandos para controlar los motores del otro robot existente en la simulación. Se debe enviar indicar distintos valores para las velocidades de los motores del robot a lo largo del tiempo. Luego fijando los motores del robot en una velocidad determinada, alguno de los agentes ejecuta el comando de cambiar el tamaño del paso de simulación, incrementándolo.

Por lo tanto el orden de la secuencia de comandos que se ejecutan es la siguiente:

- Caso de prueba 5
- Creación de entidad, tipo robot
- Especificar posición de entidad (2 veces)
- Controlar motores de entidad (varias veces)
- Especificar tamaño del paso de simulación

Para comprobar empíricamente la correctitud de este caso de prueba en parte se procede en forma análoga al caso de prueba anterior, observando los resultados en la interfaz de usuario. Se observara el cambio de posición de las entidades a las cuales se le realiza esta operación. También se observara las distintas velocidades de los motores del robot al cual se le ejecuta el comando de especificar la velocidad de los motores. Estas velocidades y el movimiento del robot se debe corresponder a lo que se espera dadas las velocidades enviadas por el agente. Una vez que se cambia el tamaño del paso de simulación, la simulación debe experimentar un cambio en su velocidad proporcional a la variación efectuada sobre este parámetro. Esta variación se puede observar y comprobar empíricamente fácilmente observando el robots en movimiento.

3.1.10. Caso 10

Este caso de prueba comienza con la ejecución del caso de prueba 5. Luego alguno de los agentes ejecuta los comandos para modificar la gravedad, la dureza global y la esponjosidad global de la simulación.

Por lo tanto el orden de la secuencia de comandos que se ejecutan es la siguiente:

- Caso de prueba 5
- Especificar gravedad
- Especificar restricciones de dureza global
- Especificar restricciones de esponjosidad global

Para comprobar empíricamente la correctitud de este caso de prueba se pueden observar diferencias en el comportamiento de los cuerpos en la simulación. Sin embargo, resulta conveniente utilizar técnicas de caja blanca para la verificación de este caso de prueba. Luego de ejecutados los comandos se realiza una consulta directamente al controlador del modelo físico para averiguar cuales son estos parámetros en ese momento. El resultado de esta consulta puede ser almacenado en un archivo o mostrado en pantalla.

3.1.11. Caso 11

Este caso de prueba comienza con la ejecución del caso de prueba 5. Luego se desea verificar el funcionamiento del simulador cambiando el modo de comunicación del sistema. Una vez inicializado el sistema se ejecuta el comando de cambiar el modo de comunicación a modo sincrónico. Sin embargo no deberían haber diferencias en el comportamiento del sistema, ya que en este caso ningún agente esta conectado en ese modo.

Por lo tanto el orden de la secuencia de comandos que se ejecutan es el siguiente:

- Caso de prueba 5
- Especificar modo de comunicación

3.1.12. Caso 12

Este caso de prueba comienza con la ejecución del caso de prueba 6. Luego se desea verificar el funcionamiento del simulador cambiando el modo de comunicación del sistema. Como existe un agente conectado de modo sincrónico, a cada paso de simulación se debería esperar un tiempo de espera determinado antes de avanzar al siguiente paso, ya que dicho agente en este caso de prueba no envía un mensaje de sincronización con determinada iteración, como debería hacerlo para notificarle al simulador que le permite avanzar la simulación.

Por lo tanto el orden de la secuencia de comandos que se ejecutan es el siguiente:

- Caso de prueba 6
- Especificar modo de comunicación
- Especificar tiempo de espera en comunicación sincrónica(varias veces)

Para comprobar la correctitud de este caso de prueba se ejecuta el comando de especificación del tiempo de espera diversas veces con diferentes valores. En cada modificación se debe observar que el lapso de tiempo entre dos avances del estado de la simulación sea el estipulado en la ejecución del comando.

3.1.13. Caso 13

Este caso de prueba comienza con la ejecución del caso de prueba 7. Luego se desea verificar el funcionamiento del simulador cambiando el modo de comunicación del sistema. Este caso debe comportarse de manera análoga al caso de prueba anterior.

Por lo tanto el orden de la secuencia de comandos que se ejecutan es el siguiente:

- Caso de prueba 7
- Especificar modo de comunicación
- Especificar tiempo de espera en comunicación sincrónica(varias veces)

3.1.14. Caso 14

Este caso de prueba comienza con la ejecución del caso de prueba 6. Luego se desea verificar el funcionamiento del simulador cambiando el modo de comunicación del sistema y sincronizando al agente con el simulador. A cada paso de iteración los agentes conectados reciben información acerca del estado del mundo. Una vez que recibe esta información, el agente que se encuentra conectado de manera sincrónica le envía al simulador una notificación expresando que ya ha procesado la información correspondiente a esa iteración y que ya es posible avanzar la simulación sin esperar. El simulador recibe esta notificación y avanza. Este ciclo se repite en todas las iteraciones.

Por lo tanto el orden de la secuencia de comandos que se ejecutan es el siguiente:

- Caso de prueba 6
- Especificar modo de comunicación
- Sincronización con iteración (a cada iteración)

Para comprobar la correctitud de este caso de prueba se puede observar el comportamiento de los objetos en la interfaz de usuario. Su movimiento debe ser fluido, ya que la respuesta del agente se produce de manera casi inmediata al recibir la información del estado del mundo a cada iteración, y el tiempo entre iteraciones nunca llega a alcanzar el tiempo de espera por defecto del simulador como en el caso de los casos de prueba anteriores.

3.1.15. Caso 15

Este caso de prueba comienza con la ejecución del caso de prueba 7. Es similar al caso de prueba anterior con la diferencia que en este caso ambos agentes están conectados de modo sincrónico y ambos agentes le envían respuestas al simulador a cada iteración, notificando le que debe avanzar. El simulador avanza una vez que ha recibido a cada iteración la respuesta de ambos agentes.

Por lo tanto el orden de la secuencia de comandos que se ejecutan es el siguiente:

- Caso de prueba 7
- Especificar modo de comunicación
- Sincronización con iteración (2 veces a cada iteración)

Para comprobar la correctitud de este caso de prueba se puede observar el comportamiento de los objetos en la interfaz de usuario. Su movimiento debe ser fluido, ya que la respuesta de los agentes se produce de manera casi inmediata, y el tiempo entre iteraciones nunca llega a alcanzar el tiempo de espera por defecto del simulador como en el caso de los casos de prueba anteriores.

3.1.16. Caso 16

Este caso de prueba comienza con la ejecución del caso de prueba 7. Es similar al caso de prueba anterior con la diferencia que uno de los agentes no envía la notificación de sincronización en cada iteración. De este modo, a pesar de que uno de los agentes le envía dicha notificación, el simulador no debe avanzar y solo lo hace cuando transcurre un tiempo igual a un tiempo de espera estipulado a partir del envió de la información de determinada iteración hacia los agentes.

Por lo tanto el orden de la secuencia de comandos que se ejecutan es el siguiente:

- Caso de prueba 7
- Especificar modo de comunicación
- Sincronización con iteración (a cada iteración)

Para comprobar la correctitud de este caso de prueba se puede observar el comportamiento de los objetos en la interfaz de usuario. El avance de la simulación debe ser muy lento ya que a cada iteración se espera un mensaje de sincronización desde ambos agentes y uno de ellos jamas lo envía.

3.1.17. Caso 17

Este caso de prueba comienza con la ejecución del caso de prueba 5. Su principal objetivo es verificar el correcto funcionamiento de las funcionalidades de persistencia y carga del sistema. Como se persiste no solo el estado de simulación, sino también el estado del sistema, se modifica un parámetro de configuración del simulador, como es el modo de paso de tiempo, para luego persistir la simulación, avanzar algunas iteraciones y cargar nuevamente de modo de intentar detectar posibles fallas.

- Caso de prueba 5
- Guardar simulación a archivo
- Cargar simulación desde archivo
- Especificar modo de paso de tiempo (modo paso a paso)
- Guardar simulación a archivo
- Especificar modo de paso de tiempo (modo normal)
- Cargar simulación desde archivo
- Especificar bandera de paso de tiempo

Lo esperado en este caso de prueba es que al cargar la simulación posteriormente a ser guardada, la misma vuelva al estado que tenia en el momento de ser guardada. Esto se puede observar a través de la interfaz de usuario del simulador. Luego de guardar la simulación y volverla a cargar luego de un lapso considerable de tiempo desde el archivo recién creado, se ejecuta el comando de especificación del modo de paso de tiempo, pasando el simulador al modo paso a paso. Después de esto se procede se realizan una serie de acciones. Primero se guarda la simulación en un archivo. Posteriormente se ejecuta nuevamente el comando de especificación de modo de paso de tiempo para volver la simulación a modo normal. De esta forma la simulación avanza normalmente y se espera un lapso considerable de tiempo para luego apreciar mejor las diferencias. Después se carga la simulación previamente guardada y se debe observar que además que el estado de los objetos de la simulación debe ser igual al que existía en

el momento de realizar la persistencia, el simulador debe estar en modo paso a paso nuevamente. Esto se puede comprobar fácilmente observando la simulación en pausa y ejecutando el comando de especificación de la bandera de paso de tiempo para avanzar la simulación en una interacción.

3.2. Resultados

3.2.1. Caso 1

El caso de prueba finalizó con éxito.

3.2.2. Caso 2

El caso de prueba finalizó con éxito.

3.2.3. Caso 3

El caso de prueba finalizó con éxito.

3.2.4. Caso 4

El caso de prueba finalizó con éxito.

3.2.5. Caso 5

El caso de prueba finalizó con éxito.

3.2.6. Caso 6

El caso de prueba finalizó con éxito.

3.2.7. Caso 7

El caso de prueba finalizó con éxito.

3.2.8. Caso 8

El caso de prueba finalizó con éxito.

3.2.9. Caso 9

El caso de prueba finalizó con éxito.

3.2.10. Caso 10

El caso de prueba finalizó con éxito.

3.2.11. Caso 11

El caso de prueba finalizó con éxito.

3.2.12. Caso 12

El caso de prueba finalizó con éxito.

3.2.13. Caso 13

El caso de prueba finalizó con éxito.

3.2.14. Caso 14

El caso de prueba finalizó con éxito.

3.2.15. Caso 15

El caso de prueba finalizó con éxito.

3.2.16. Caso 16

El caso de prueba finalizó con éxito.

3.2.17. Caso 17

El caso de prueba finalizó con éxito.

4. Pruebas no funcionales

Se desea comprobar el correcto funcionamiento del simulador sometido a una gran carga de trabajo. Para esto se crearan escenarios en los que sucesivamente se incrementaran la cantidad de entidades de simulación, los agentes conectados y el trafico de mensajes generado. La métrica que se utilizara para observar el degradamiento de la performance a medida que aumenta la carga será el lapso de tiempo promedio entre dos iteraciones del simulador. Cada caso de prueba se ejecutará una vez con interfaz gráfica y otra sin ella. Esto se hará para comparar ambos resultados y obtener información que indique en que grado afecta la interfaz gráfica al rendimiento del simulador. En todos los casos el simulador y los agentes estarán ejecutándose en modo sincrónico. Los agentes se ejecutan en la misma computadora que el simulador, algo que sin duda influye en el deterioro del rendimiento a medida que aumenta la carga. Las pruebas serán realizadas en un sistema con procesador AMD Sempron 2200 y 128MB de memoria RAM.

4.1. Casos de prueba

4.1.1. Caso 1

En este caso de prueba se conectarán al simulador 5 agentes cada uno de los cuales creará una entidad de tipo robot y alguno de ellos creará una entidad de tipo pelota y otra de tipo campo de juego. Todos los agentes ejecutarán comandos para controlar motores de uno de los robots y enviarán esta información en cada iteración.

4.1.2. Caso 2

En este caso de prueba es análogo al anterior pero se ejecutará por un largo periodo de tiempo, si el rendimiento se deteriora a media que avanza el tiempo esto puede indicar un incorrecto manejo de memoria.

4.1.3. Caso 3

En este caso de prueba se conectarán al simulador 1 agente que creará 10 entidades de tipo robot, una entidad de tipo pelota y otra de tipo campo de juego. Este agente ejecutará comandos para controlar motores de todos los robots y enviará esta información en cada iteración.

4.1.4. Caso 4

En este caso de prueba se conectarán al simulador 5 agentes cada uno de los cuales creará dos entidades de tipo robot y alguno de ellos creará una entidad de tipo pelota y otra de tipo campo de juego. Todos los agentes ejecutarán comandos para controlar motores de dos de los robots y enviarán esta información en cada iteración.

4.1.5. Caso 5

En este caso de prueba se conectarán al simulador 10 agentes cada uno de los cuales creará una entidad de tipo robot y alguno de ellos creará una entidad de tipo pelota y otra de tipo campo de juego. Todos los agentes ejecutarán comandos para controlar motores de uno de los robots y enviarán esta información en cada iteración.

4.1.6. Caso 6

En este caso de prueba se conectarán al simulador 20 agentes cada uno de los cuales creará una entidad de tipo robot y alguno de ellos creará una entidad de tipo pelota y otra de tipo campo de juego. Todos los agentes ejecutarán comandos para controlar motores de uno de los robots y enviarán esta información en cada iteración.

4.2. Resultados

4.2.1. Caso 1

En este caso el tiempo medio de iteración fue de 25.3 ms y 14.1 ms con y sin interfaz gráfica respectivamente. Se puede afirmar que en este caso de prueba el tiempo necesario para mantener la interfaz gráfica corresponde al 43.8 % del tiempo total de iteración.

4.2.2. Caso 2

En este caso el tiempo medio de iteración fue de 27.8 ms y 15.0 ms con y sin interfaz gráfica respectivamente. Se puede afirmar que en este caso de prueba el tiempo necesario para mantener la interfaz gráfica corresponde al 46 % del tiempo total de iteración.

4.2.3. Caso 3

En este caso el tiempo medio de iteración fue de 28.9 ms y 15.7 ms con y sin interfaz gráfica respectivamente. Se puede afirmar que en este caso de prueba el tiempo necesario para mantener la interfaz gráfica corresponde al 45.6 % del tiempo total de iteración.

4.2.4. Caso 4

En este caso el tiempo medio de iteración fue de 39.9 ms y 27.7 ms con y sin interfaz gráfica respectivamente. Se puede afirmar que en este caso de prueba el tiempo necesario para mantener la interfaz gráfica corresponde al 30.6 % del tiempo total de iteración.

Caso de prueba	Con GUI	Sin GUI
Caso 1	25.3 ms	14.1 ms
Caso 2	27.8 ms	15.0 ms
Caso 3	28.9 ms	15.7 ms
Caso 4	39.9 ms	27.7 ms
Caso 5	61.0 ms	50.2 ms
Caso 6	144.5 ms	128.3 ms

Cuadro 1: Comparación de resultados

4.2.5. Caso 5

En este caso el tiempo medio de iteración fue de 61.0 ms y 50.2 ms con y sin interfaz gráfica respectivamente. Se puede afirmar que en este caso de prueba el tiempo necesario para mantener la interfaz gráfica corresponde al 18 % del tiempo total de iteración.

4.2.6. Caso 6

En este caso el tiempo medio de iteración fue de 144.5 ms y 128.3 ms con y sin interfaz gráfica respectivamente. Se puede afirmar que en este caso de prueba el tiempo necesario para mantener la interfaz gráfica corresponde al 11.2 % del tiempo total de iteración.

4.2.7. Comparación

Se puede concluir que el factor que mas afecta al rendimiento del sistema es la cantidad de agentes conectados al mismo. Sin embargo hay que tener en cuenta que en las pruebas realizadas todos los agentes se encontraban ejecutándose en la misma computadora. Esto implica que en el ultimo caso de de prueba se ejecutaron 21 procesos (el simulador y cada uno de los 20 agentes). Cada agente generaba 2 mensajes por iteración, uno para sincronizarse con la iteración y otro para controlar los motores de un robot. Además el simulador realizaba broadcast del estado del mundo a cada iteración, enviándole dicha información a cada agente. Por lo tanto, se generaba un trafico de 60 mensajes diferentes por iteración que debían ser decodificados y procesados, lo cual explica el deterioro de performance. Existen diversas vías para mejorar sensiblemente la performance del sistema, pero la idea de estas pruebas era justamente someter al sistema a una gran carga de trabajo en lo que concierne al procesamiento de mensajes.

Por otra parte se puede observar que el peso relativo de mantener la interfaz gráfica con respecto al procesamiento total del sistema, se ve sensiblemente atenuado a medida que aumenta la carga de trabajo.

A continuación se presenta una tabla que resume los resultados obtenidos.

5. Conclusión

Las pruebas realizadas al sistema presentaron un resultado satisfactorio. Todos los casos de pruebas funcionales planteados fueron ejecutados con éxito. Entre los casos de prueba no funcionales se puede observar que el deterioro en el rendimiento está asociado al incremento en el numero de agentes y tráfico de mensajes. No se realizaron pruebas con mensajes mal formados, o sea, si un agente envía mensajes que no cumplen los requisitos sintácticos que se espera de los mensajes, no se garantiza que el sistema siga funcionando normalmente. Aunque el sistema presenta diversos mecanismos para ajustar el rendimiento del simulador a la realidad planteada y los tiempos registrados pueden mejorarse notablemente, está claro que uno de los puntos a tener en cuenta en la optimización del sistema es el procesamiento de mensajes, hacerlo de una manera más eficiente puede contribuir enormemente para incrementar el rendimiento del sistema. Además se observó un lento deterioro del rendimiento a medida que pasa el tiempo, por lo que esto puede indicar un mal manejo de memoria, por lo tanto realizar una revisión de esto sería importante.