

Guía 2: Fundamentos del control de Robots

Esta sesión permitirá explorar los aspectos fundamentales involucrados con el movimiento de robots en el contexto de RoboSoccer, indagando en los conceptos de movimientos, giro, pase, y patear el balón. Aprendiendo como programar estas acciones directamente a los agentes en el campo, con el fin de eliminar la necesidad de ingresar estos comandos por la aplicación de la terminal.

1.1 Equipo Helios-base

Este algoritmo, escrito en el lenguaje de programación C++, corresponde a un equipo de muestra para el servidor de simulación en dos dimensiones de la RoboCup.

Desarrollar un programa de agente desde cero es un difícil desafío, ya que se necesitan una gran cantidad de módulos complejos, como una comunicación de red estable, sincronización, modelado de mundo, entre otros, para producir un algoritmo de agente que funcione completamente en el simulador. Es por ello por lo que se presenta un equipo básico, que pueda permitirles a los interesados en participar de esta disciplina concentrarse en técnicas de trabajo en equipo.

Con el fin de comprender como funciona un equipo de RoboSoccer profesional se instalará el equipo Helios-base. Las instrucciones de instalación se encuentran a continuación:

Paso 1:

Descargar el fichero correspondiente a través del siguiente enlace.

- <https://github.com/helios-base/helios-base.git>

O desde la terminal usando el comando:

```
1 git clone https://github.com/helios-base/helios-base.git
```

Esta carpeta debe encontrarse en el directorio principal del usuario (home/"usuario"), con el fin de posicionarla en una dirección a la que sea fácil navegar.

Paso 2:

Es necesario ejecutar los siguientes comandos en la terminal para asegurar el entorno de desarrollo básico:

```
1 sudo apt update
2 sudo apt install build-essential libboost-all-dev
```

Paso 3:

Ingresar los siguientes comandos en la terminal en el orden establecido:

```

1 sudo unzip helios-base-master.zip
2 cd helios-base-master
3 ./bootstrap
4 ./configure
5 sudo make

```

Este proceso finaliza la instalación del equipo Helios-base. Para ejecutar este algoritmo es necesario moverse al fichero src, el cual se encuentra dentro de la carpeta principal del equipo, esto se puede lograr ingresando el siguiente comando:

```

1 cd ~ //Este comando se utiliza para volver al directorio principal del usuario.
2 cd helios-base-master/src

```

Luego solo se debe ejecutar el siguiente comando para inicializar el equipo. Cabe destacar que para que esto sea exitoso el servidor de RoboSoccer debe estar activo en el punto previo a la patada inicial.

```

1 ./start.sh

```

Si se desea inicializar a un equipo contrincante usando este mismo algoritmo se debe abrir una segunda terminal e ingresar el siguiente comando, reemplazando la variable "Team" por el nombre deseado para el segundo equipo.

```

1 ./start.sh -t Team

```

La siguiente figura muestra la interfaz de monitor una vez inicializados ambos equipos.

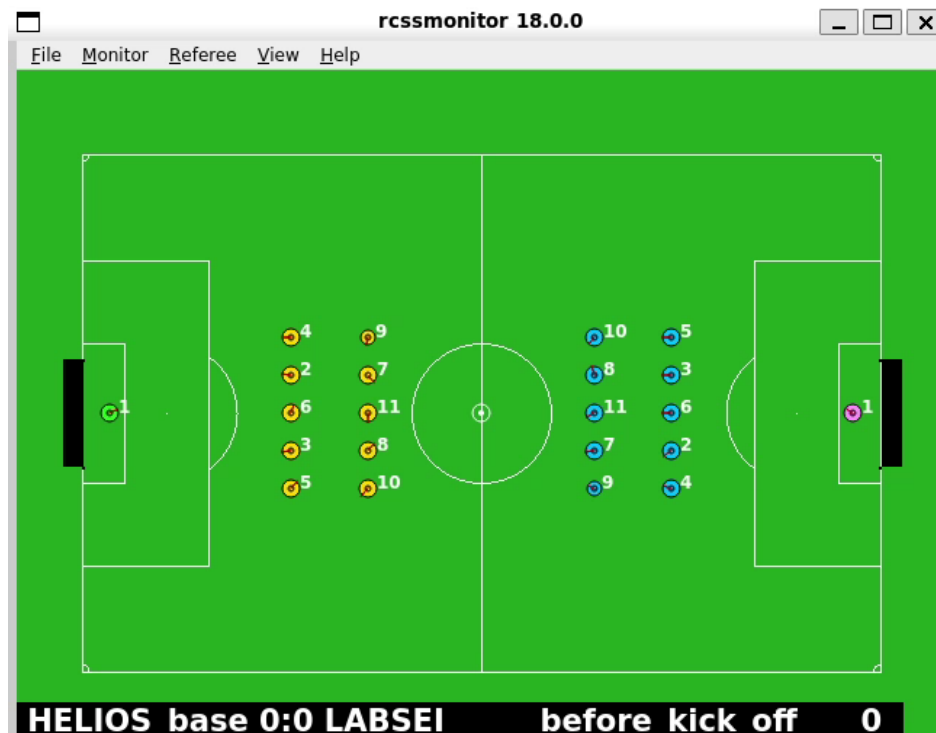


Figura 1: Simulación previo al inicio del partido.

Con el fin de familiarizarse con la dinámica de un partido se requiere como actividad previa, el seguir los pasos anteriores y observar un partido de soccer simulado utilizando el equipo básico Helios.

1.2 Comandos de movimiento de los agentes

Los clientes (jugadores) disponen de la facultad de implementar comandos de movimiento, comunicación y peticiones específicas, los cuales se encuentran detallados en la siguiente tabla, en conjunto con la explicación de su función dentro del servidor:

Tabla 1: Comandos de movimiento del cliente

Comando	Función
(turn <i>Momento</i>)	El <i>Momento</i> se expresa en grados de -180 a 180. Este comando gira la dirección del cuerpo del jugador según los grados ingresados con respecto a la dirección actual.
(dash <i>Potencia</i>)	Acelera al jugador en la dirección de su cuerpo (no en la dirección de la velocidad actual). La potencia se encuentra dentro de un rango definido por el servidor, con valores máximos (maxpower) y mínimos (minpower) establecidos (por defecto, -100 y 100).
(kick <i>Potencia Direccion</i>)	Acelera el balón con la Potencia y Dirección ingresada. La dirección es relativa a la dirección del cuerpo del jugador y la potencia se encuentra nuevamente entre minpower y maxpower.
(catch <i>Direccion</i>)	Comando especial utilizado exclusivamente por el guardametas. . Tiene como objetivo principal atrapar el balón. La dirección indicada en el comando se refiere a la posición relativa al cuerpo del portero. Si la acción de atrapar es exitosa, el portero mantendrá el balón en sus manos hasta que decida patearlo.
(move X Y)	Este comando solo puede ser ejecutado antes del saque inicial y después de un gol. Su función es mover al jugador a la posición exacta de X (entre -54 y 54) e Y (entre -32 y 32) en un ciclo de simulación. Esto es útil para arreglos antes del saque inicial.
(turn_neck <i>Angulo</i>)	Este comando permite girar el cuello del jugador según el ángulo indicado con relación al ángulo previo. Es importante destacar que el ángulo resultante del cuello estará comprendido entre minneckang (por defecto: -90) y maxneckang (por defecto: 90) con respecto a la dirección del cuerpo del jugador.
(sense_body)	Solicita al servidor que envíe información del cuerpo sensorial. Es importante tener en cuenta que el servidor envía esta información en cada ciclo.
(score)	Solicita al servidor que envíe información sobre el marcador.

Cada partido se divide en 6000 ciclos, y cada ciclo tiene una duración de 100 ms. Estos ciclos se dividen en dos mitades de 3000 ciclos cada una, equivalente a una duración de 5 minutos por mitad. Durante este

tiempo limitado, los agentes deben analizar la información recibida, calcular una representación precisa y libre de ruido y determinar la acción más adecuada a tomar.

2 Actividad 1:

Duración aproximada: 30 minutos.

En la siguiente actividad pondremos en práctica los conocimientos básicos de movimiento de un jugador, con el fin de programar un agente guardametas, cuyo deber consista en mantenerse en el área de meta, con el fin de proteger la portería de los disparos del equipo contrario.

Los objetivos de esta actividad son:

- Familiarizarse con el equipo de RoboSoccer básico Helios y su funcionamiento
- Inicializar correctamente al jugador guardametas (goalie).
- Diseñar un algoritmo que habilite el movimiento del guardametas en la portería.
- Asegurar la protección de la portería mediante el desplazamiento adecuado del guardametas.

Para completar esta actividad es necesario inicializar a un jugador guardametas, esto es posible alterando ligeramente el mensaje de saludo utilizado en la Guía 1, al incluir la variable (goalie). El siguiente comando muestra un ejemplo de este saludo:

```
1 (init NombreEquipo (goalie))
```

Luego de inicializar a este agente en el campo es necesario diseñar un algoritmo que le permita moverse de forma adecuada. Para luego, restringir su movimiento al área de la portería. Para llevar a cabo esta tarea se recomienda analizar los algoritmos del equipo base Helios Base. Específicamente los códigos titulados: "start.sh", este código representa el algoritmo principal del equipo, siendo posible indicar la cantidad de jugadores y nombre de su equipo. El código "bhv_goalie_basic_move.cpp" que representa los movimientos básicos de un guardametas. Y por último, los algoritmos "role_goalie.cpp" y "strategy.cpp", los cuales inicializan el rol de un agente guardametas en el campo.

Al completar esta actividad, se espera haber logrado los siguientes resultados:

- El agente identificado como guardametas es iniciado en el campo de juego.
- El agente guardametas es capaz de desplazarse por el área de juego.
- El agente guardametas se mantiene dentro del área de la portería.

La Figura 2 ilustra los resultados esperados:



Figura 2: Jugador guardametas iniciado en el campo de juego.

Esta actividad busca adaptar los conocimientos de movimiento de un jugador, para restringirlo a un área específica. El programar un guardametas eficiente contribuye al desempeño defensivo del equipo y aumenta las posibilidades de éxito en los partidos.

3 Actividad 2:

Duración aproximada: 45 minutos.

En la siguiente actividad los participantes desarrollaran habilidades prácticas de programación y control de robots, diseñando estrategias de movimiento y comunicación con el fin de completar acciones específicas en un entorno de simulación.

Los objetivos de esta actividad son:

- Desarrollar habilidades de programación para crear un algoritmo de cliente.
- Aprender a establecer una comunicación efectiva entre el jugador y el servidor.
- Practicar el control de movimiento del jugador.
- Implementar un jugador ofensivo y un agente guardametas de forma simultánea.

Esta actividad consiste en crear un algoritmo de cliente, basándose en el equipo de RoboSoccer básico Helios-base, el cual debe permitir que un jugador se comuniqué con el servidor, y pueda moverse hacia una ubicación específica, para luego patear el balón hacia un objetivo.

Para completar esta actividad es necesario en primer lugar crear un algoritmo capaz de enviar el comando de inicialización para dos jugadores. Luego de completar, el paso anterior es necesario analizar el contenido de la carpeta "src", del equipo Helios, este fichero contiene todos los algoritmos que controlan a cada jugador del equipo base, es de especial importancia el algoritmo llamado "bhv_basic_movement". Luego de completar la inicialización de los agentes en el campo, es necesario diseñar un algoritmo de movimiento que busque posicionar a un jugador en un punto específico del campo. Finalmente, el jugador

se debe desplazar al centro del campo y patear el balón, el cual, siempre se encuentra en la posición central del campo al momento de iniciar el juego.

Al completar esta actividad, se espera haber logrado los siguientes resultados:

- Un algoritmo de cliente que permita iniciar a dos jugadores en el campo.
- El jugador debe ser capaz de moverse hacia una ubicación objetivo en el campo.
- El jugador debe ser capaz de patear el balón.

La Figura 3 ilustra los resultados esperados:



Figura 3: Jugador luego de patear el balón.

Esta actividad es fundamental para desarrollar habilidades de programación y control en el contexto de RoboSoccer. Al completar esta actividad, los participantes adquirirán experiencia en el establecimiento de comunicación con el servidor, el control de movimiento del jugador y la implementación de acciones específicas. Formando así las bases para la creación de un equipo.

Referencias

- [1] Akiyama, H. (s.f.). *HELIOS Base: A base team for the RoboCup Soccer Simulation*. Recuperado el 28 de Junio de 2023, de <https://github.com/helios-base>
- [2] Akiyama, H., & Nakashima, T. (2014). HELIOS Base: An Open Source Package for the RoboCup Soccer 2D Simulation. *RoboCup2013: Robot World XVII, Lecture Notes in Artificial Intelligence*. Berling: In Sven Behnke, Manuela Veloso, Arnoud Visser, and Rong Xiong editors.
- [3] RoboCup Federation. (s.f.). *RoboCup Federation official website*. (RoboCup Federation) Recuperado el 2 de Julio de 2023, de <https://robocup.org>
- [4] The RoboCup Soccer Simulator Maintenance Committee. (s.f.). *The robocup soccer simulator users manual*. Recuperado el 02 de Julio de 2023, de <https://rscsoccersim.readthedocs.io>