

Un agente es todo aquello que puede ser visto como percibiendo su ambiente por medio de sensores y actuando sobre ese ambiente mediante actuadores.

Internamente, el comportamiento de un agente está implementado por un programa agente (signo de interrogación) que determina qué acción ejecutar. En el contexto de T-World los agentes se

Én el contexto de T-World los agentes se corresponden con los robots de la simulación 3D y los programas agente con algoritmos escritos en cualquier lenguaje de programación.

WebWorker Gogs WebWorker JunaScript T-World

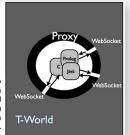
Programas Agente

Caso JavaScript

El algoritmo corre dentro de un WebWorker, e.d. en un script en el background independiente del script principal que atiende la simulación 3D

Otro caso

El algoritmo corre en un programa escrito en un lenguaje externo al navegador (ejemplo, C, Prolog o Java) y requieren del Servidor Proxy para comunicarse con T-World.

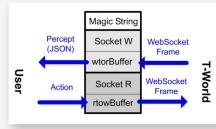


T-World (navegador web)



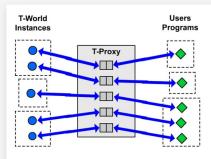
(programas de usuarios)

El proxy se encarga de enlazar cada agente (robot) de la simulación 3D con su respectivo programa agente. Dado que la simulación transcurre dentro del navegador web, T-World asigna un WebSocket a cada robot y conecta cada WebSocket a su respectivo proxy. Es por esto que el diálogo entre T-World y el proxy se hace por medio del protocolo WebSocket. Por otra parte, el programa del usuario (programa agente) sólo debe conectarse al proxy, recibir las percepciones en JSON (o XML) desde T-World y enviar las correspondientes acciones.



Estructura dedicada para cada par de conexión (robot programa de usuario)

- conexión (robot, programa de usuario).
 Magic String: almacena identificador bajo el cual están apareados los sockets.
- SocketR y SocketW: almacenan el socket de usuario y el WebSocket.
- wtorBuffer y rtowBuffer: buffers de 16KB utilizados para el reenvio de mensajes desde el WebSocket hacia el usuario y viceversa.



La estructura superior le da al proxy la capacidad de soportar múltiples pares de conexiones, todos ellos conectados de manera concurrente al proxy.

Dentro del proxy cada estructura mantiene un par de conexiones. Los circulos representan WebSockets, y el cuadrado donde éstos se encuentran instancias de T-World. Los rombos representan sockets de usuarios, cada cuadrado contenedor representa una instancia de programa de agente (de usuario).

Plataforma Web T-World

Autor: Sergio Burdisso (sergio.burisso@gmail.com)
Asesores: Guillermo Aguirre, Marcelo Errecalde
UNSL

Introducción

T-World (*tworld-ai.com*) es una plataforma web que permite crear distintas configuraciones de un entorno en 3D para evaluar el comportamiento de agentes inteligentes. Para la investigación y docencia en Inteligencia Artificial es importante contar con tareas de referencia para analizar, experimentar y entender las particularidades que surgen al combinar distintos tipos de agentes con distintos tipos de entornos. Considerando que el desarrollo de los agentes se realiza en distintas plataformas y haciendo uso de diversos lenguajes de programación fue necesario contar con un programa intermediario que los enlace con T-World. Por esa razón T-World incorpora un Proxy que permite a un programa escrito en cualquier lenguaje de programación conectarse con la simulación 3D dentro del navegador web. Además, en caso de que se desee escribir el programa agente en JavaScript, T-World brinda un editor de código online y una API para simplificar y acelerar el proceso de codificación del mismo.



Conclusiones

Los objetivos que se plantearon desde el inicio del diseño de T-World, fueron variados, entre otros que fuera accesible desde cualquier parte del mundo, de manera que simplemente ingresando a un sitio web se tenga acceso a una herramienta completa e integrada de forma nativa al navegador web. Que su funcionamiento fuese independiente de un sistema operativo en particular o de cualquier otra pieza de software -como interpretes, runtime frameworks o maquinas virtuales. Para lograr estos objetivos, entre otras cosas, fue requerido codificar más de 18 mil líneas de código - 12 mil en JavaScript, aproximadamente mil en Lenguaje C y el resto en HTML y PHP. También fue necesaria la utilización de diversas tecnologías tales como Angular JS de Google como framework para la aplicación web y la biblioteca CopperLicht como una abstracción de WebGL para la generación de gráficos 3D. Cabe mencionar que, como parte del trabajo, se tuvieron que realizar cambios y mejoras significativas a Copperlicht, estos cambios fueron notificados a los autores originales y posteriormente añadidos a la biblioteca oficial.