RoboCupJunior Rescue New Simulation 2021

Team Description Paper

Titán

## Abstract

1. Nuestro robot es capaz de generar una representación del mapa de la competencia y encontrar una secuencia de movimientos óptima para su navegación. Nuestro código tiene una arquitectura organizada y hemos desarrollado un conjunto de herramientas para facilitar la programación a alto nivel. Todo nuestro trabajo está documentado en github.

## Introduction

## Team

* + 1. Nuestro equipo está formado por Máximo Cansino, Lucas Flores y Alejandro de Ugarriza. Alejandro participó con otro equipo en la Primera Demonstración Simulada, Máximo y Alejandro fueron ganadores junto con otro compañero de la Roboliga Simulada en Argentina realizada en la plataforma de Erebus y Lucas es una nueva adición al equipo. Máximo se encarga de la detección de víctimas, Alejandro de mapeo y navegación y Lucas de la grilla entregable para finalizar el juego.

## Aim, strategy, and overall plan

* + 1. Nuestro objetivo es hacer un robot capás de navegar inteligentemente, analizar y mapear cualquier tipo de entorno. Calcular trayectorias optimas empleando complejos algoritmos para una navegación eficiente. Detectar víctimas y carteles empleando estrategias simples procesamiento de imágenes. Hacer un código más organizado, responsable, robusto y documentado a través de GitHub.
    2. Nuestra estrategia como equipo consiste principalmente en la división y asignación de roles para cada uno de los integrantes, coordinándonos a través de GithHub y reuniones virtuales una vez a la semana. Realizar una nueva arquitectura para el código y luego trasladar y modificar pedazos de nuestro código anterior allí, además de crear nuevas soluciones más robustas y eficientes.
    3. Empezamos realizando un esquema para la nueva arquitectura, definimos las clases en el código, luego empezamos a trasladar pedazos del código anterior, optimizándolos y adaptándolos a los cambios requeridos por la nueva plataforma. Continuamos definiendo el funcionamiento de sensores y actuadores básicos, luego los movimientos de precisión en el mapa, implementamos el mapeo del laberinto, luego los algoritmos de pathfinding y luego integramos todo y definimos la lógica de navegación. Paralelamente desarrollamos un sistema de detección de víctimas y carteles.

## Technical progress

## Navigation

* + 1. Nuestro Robot navega moviéndose en líneas rectas de vértice en vértice, un vértice siendo considerado a la escala de medias baldosas. Es capás de realizar secuencias de movimientos pre-programadas en situaciones clave, y estas son muy fáciles de programar (Referirse a la sección de máquinas de estado).

Nuestro sistema de navegación está integrado por dos secciones: Análisis y Movimientos de bajo nivel.

* Análisis se encarga de planear la navegación a largo plazo para recorrer más efectivamente el mapa, y no tiene ninguna conexión directa con los actuadores. Utiliza entre otras cosas, breadth first search para encontrar la mejor casilla posible para moverse en la representación virtual del mapa. Una vez encontrada utiliza A Star para encontrar el camino mas corto para llegar a esta posición. Está encapsulado en la clase Analysis.
* La parte encargada de movimientos de bajo nivel tiene como función controlar los actuadores directamente para seguir el camino proveniente de Análisis o para sacar al robot de una emergencia que requiera control inmediato y directo. Esta distribuido entre la clase de RobotLayer y AbstractionLayer.

Ahora mismo tenemos únicamente los movimientos a bajo nivel en el nuevo código que desarrollamos para esta competencia, aunque tenemos gran parte de Análisis hecha en códigos anteriores, pero sin implementar.

* + 1. Dividimos la navegación en corto y largo plazo. Utilizamos la diferencia en posiciones del gps a través del tiempo para obtener una lectura confiable de la rotación global, esto nos permite recalibrarla en cualquier momento independientemente de la teletransportación o errores en las medidas del gyro.
    2. Para finalizar solo falta adaptar e implementar la parte de Análisis de códigos anteriores. En un futuro planeamos implementar un sistema más complejo para encontrar las casillas óptimas para movernos y optimizar el movimiento a corto plazo para lograr más rapidez y fluidez. A muy largo plazo quizás incluyamos herramientas de Inteligencia Artificial más complejas e incluso Machine Learning.

## Detection

* + 1. Explanation of your victim/hazard map detection system. What it is, what(and how it) works, what doesn’t work. Include any innovative features.
    2. Talk about particularly innovative or new technologies/strategies your team used to solve this problem.
    3. How you plan to solve the things that don’t work currently. What you want to implement in addition to the current system, etc.

## Mapping

* + 1. Para mapear utilizamos el sensor LIDAR y el sensor de color. Los datos de estos son procesados y mandados junto con la posición y rotación del robot. Luego las detecciones del LIDAR son comparados a distintas “plantillas” para determinar que representan y son representados en una grilla. Las plantillas nos dan control completo sobre el sistema de mapeo. En la grilla están representados cada baldosa (media baldosa, 0.06 x 0.06), vértice y pared como objetos con diferentes propiedades. La grilla tiene la capacidad de ampliarse dinámicamente a medida que crece el mapa.
    2. Talk about particularly innovative or new technologies/strategies your team used to solve this problem.
    3. How you plan to solve the things that don’t work currently. What you want to implement in addition to the current system, etc.

## Hardware

* + 1. Anything else which is relevant to your team but was not covered by the above sections.

## Program flow and sequential code execution:

* + 1. Desarrollamos un sistema para poder controlar en Flow de nuestro programa en forma de una máquina de estados, además de una manera de insertar pedazos de código secuencial al programa, donde se pueden usar, entre otras cosas, delays sin afectar otras partes del código que deben ejecutarse continuamente. Esto nos permite hacer pruebas y secuencias predeterminadas muy fácilmente.

## Performance evaluation

* 1. Evaluate your performance of your robot.

## Conclusion

* 1. Como conclusión, mejoramos nuestro sistema Teniendo en cuenta la experiencia de la competencia pasada,

## Appendix

N.B.: The appendix is NOT to continue writing the main text. It should be reserved for additional info if the reader is interested or curious to know more. Teams may link a link to external documentation as an alternative to the appendix.

## Reference