



RoboCup
с 10 по 11 ноября 2024



ROBOCUPJUNIOR RESCUE SIMULATION 2024

ТЕХНИЧЕСКИЙ ЖУРНАЛ КОМАНДЫ

кВАНт

Аннотация

Команда «кВАНт» представляет описание алгоритмов и робота для участия в RoboCupJunior Rescue Simulation 2024. Наш робот — модель от Erebos с подключёнными датчиками и моторами, которые позволяют ему исследовать лабиринт, искать и распознавать «метки на стенах». Алгоритм для микроконтроллера написан на языке программирования C++ и обеспечивает безопасное прохождение поля роботом, объезд сложных препятствий, и определение координат найденных им знаков.

Введение

Команда

Состав команды «кВАНт» Областного государственного бюджетного образовательного учреждения «Томский физико-технический лицей»:

- Гужихин Иван (программирование, изучение работы датчиков Webots, работа над распознаванием «меток на стенках», создание технического описания);
- Бан Ангелина (программирование, создание тренировочного лабиринта, изучение работы датчиков Webots, создание плаката);
- Наш тренер: Косаченко Сергей Викторович

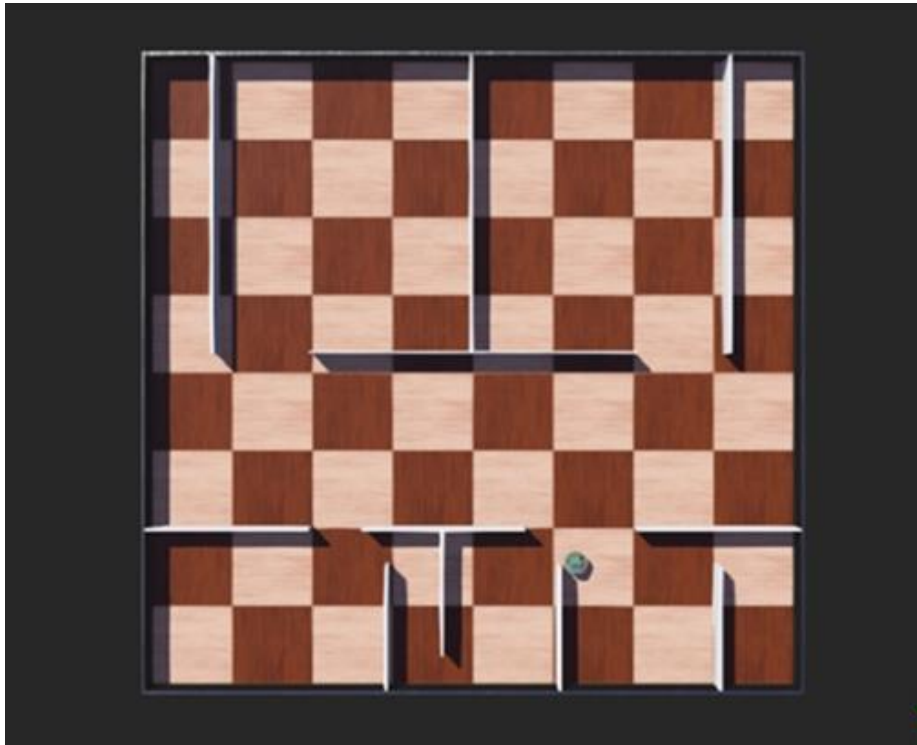
Опыт участия в робототехнических соревнованиях:

- Диплом за 1 место, XII Региональная олимпиада по образовательной робототехнике школьников Томской области 2024 в регламенте RoboCupJunior Rescue Simulation.
- Диплом за 3 место, Кубок Губернатора Томской области по образовательной робототехнике для детей, регламент «Состязания роботов с техническим зрением памяти Виктора Ширшина», 2023г. (команда «кВАНт»)
- Диплом победителя, Всероссийская ярмарка технологических проектов «Expo-TECH Junior», номинация «Робототехнические и транспортные системы», проект «Работоспособный прототип устройства «Умная теплица «TERRABOT»», 2023г.
<https://kvantoriumtomsk.ru/tpost/cx6dfriidu1-kvantorianets-iz-parabeli-stal-pobeditel> (Иван Гужихин)
- Диплом за 3 место, Кубок Губернатора Томской области по образовательной робототехнике для детей, регламент «Лабиринт туда и обратно», 2022г. https://vk.com/wall-112410914_656 (Иван Гужихин)

- Дипломы за 1 место, Кубок Губернатора Томской области по образовательной робототехнике для детей, 2018г., 2019г. (Иван Гужихин)
- Участник прямого отбора на РобоФинист (Иван Гужихин)
- Диплом 1 степени, Городская программа воспитания и дополнительного образования «Образовательная робототехника» соревнования «Осенний кубок», 2023г. (Ангелина Бан)
- Диплом за 3 место, IX Межрегиональный открытый фестиваль научно-технического творчества «Робоарт-2023», соревнования «Эстафета», 2023г. (Ангелина Бан)
- Диплом за 3 место, Открытые соревнования по робототехнике «РобоСеверск-Х», регламент «Сортировка», 2023г. (Ангелина Бан)
- Диплом 2 степени, Городская программа воспитания и дополнительного образования «Кубок Робомороза» соревнования «Осенний кубок», 2023г. (Ангелина Бан)
- Диплом 1 степени, Городская программа воспитания и дополнительного образования «Образовательная робототехника» соревнования «Осенний кубок», 2022г. (Ангелина Бан)
- Диплом за 2 место, VII Соревнования по образовательной робототехнике на Кубок Губернатора Томской области для детей, состязание Марафон шагающих роботов, 2021г. (Ангелина Бан)

План проекта

- Цель: создание алгоритма управления автономного робота для поиска и идентификации “буквенных жертв” и предупредительных “знаков опасности” на смоделированном поле за время не более 8 минут.
- План:
 - Освоение симулятора Webots
 - Создание тренировочного поля (рис.1)
 - Отладка алгоритма перемещения по лабиринту на модели робота e-ruck
 - Установка Erebus и изучение его возможностей
 - Размещение датчиков Erebus в утилите Erebus Robot Customisation и оформление файла .json подключение их в программе
 - Написание алгоритма для распознавания “меток на стенах”
 - Доработка алгоритма управления робота



Тренировочное поле. Рис.1

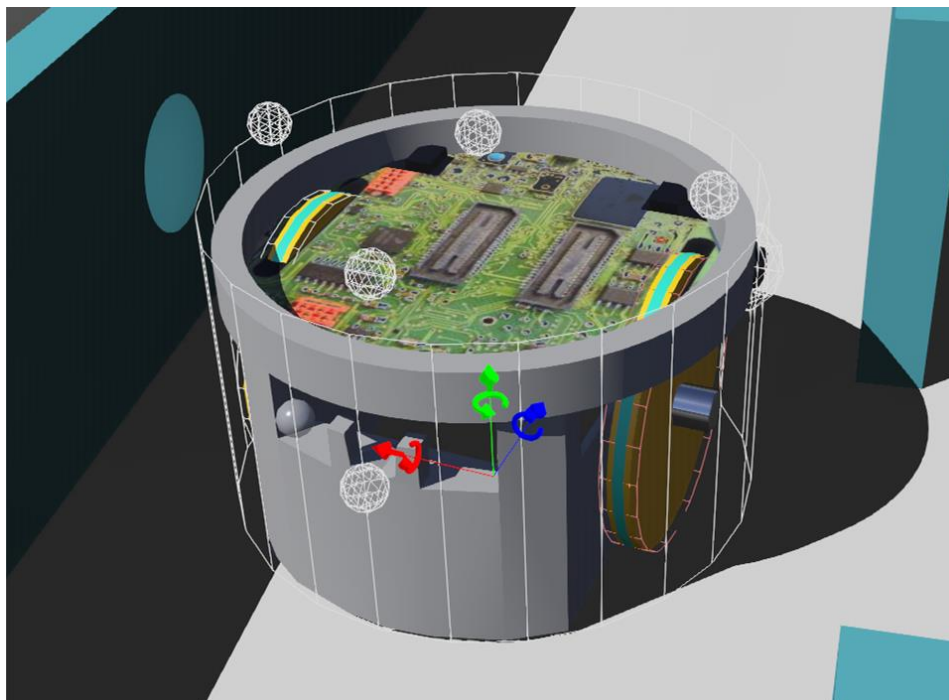
- Во время работы мы столкнулись с несколькими проблемами, например, в некоторых случаях робот мог остановиться на стенке, стоящей торцом к роботу, так как робот иногда поворачивал не под прямым углом. Для решения этой проблемы, мы добавили гироскоп, чтобы было возможно фиксировать угол поворота.
- Также накладываются ограничения на использование других циклов while, for и рекурсивных функций, так как основным циклом симулятора является цикл while (robot->step(timeStep) != -1) и симулятор требует вернуться к началу timeStep за один внутренний тик. Для решения данной проблемы нам пришлось скорректировать алгоритм без использования внутренних циклов.
- Появлялась ошибка, если помимо main мы прописывали какие-либо другие аналогичные функции (например, отвечающие за распознавание)
- Составляло сложность определиться со способом распознавания меток (алгоритм бинаризации и поиска контуров не всегда работает стабильно). Мы рассмотрели несколько вариантов, которые применимы в библиотеке OpenCV: функцию MatchTemplate() и структуру Blob – но с помощью первой невозможно обрабатывать видеопоток и определять меру идентичности шаблона, а Blob не позволяет вычислять вытянутые пятна сложной формы (например, букву “S” – одну из “жертв”).

Конструкция робота

- Наш робот обладает множеством различных датчиков, помогающих эффективно исследовать карту. (рис.2)

Sensors:	Distance Sensors (50):	Wheels (300):
<input checked="" type="checkbox"/> Gyro (100) ▼	Number of Distance Sensors 0	Number of Wheels 0
<input checked="" type="checkbox"/> InertialUnit (100) ▼	Distance Sensor 1 ▼	Wheel 1 ▼
<input checked="" type="checkbox"/> GPS (250) ▼	Distance Sensor 2 ▼	Wheel 2 ▼
<input checked="" type="checkbox"/> Camera (500) ▼	Distance Sensor 3 ▼	
<input checked="" type="checkbox"/> Camera (500) ▼		
<input type="checkbox"/> Camera (500)		
<input checked="" type="checkbox"/> Colour sensor (100) ^ X: 0 284 Y: 0 110 Z: 0 221 Angle-Axis Rotation: X: -1 0 0 Angle: 1.57 Euler Rotation: X: -1.57 0 0 Name: <input type="text" value="colour_sensor"/>	Distance Sensor 4 ^ X: 0 Y: 70 Z: -370 Angle-Axis Rotation: X: -0.68 0.68 0.28 Angle: 2.6 Euler Rotation: X: -1.57 -0.79 1.57 Name: <input type="text" value="distance sensor4"/> When using this sensor from your controller, the following tag name must be specified in your code: distance sensor4	

Датчики и моторы нашего робота. Рис.2



В левой части картинки – подсветка от ColorSensor, перед роботом дополнительный DistanceSensor (рёбра выделены белым цветом). Рис.3

Особенностью нашего робота является наличие дополнительного датчика расстояния DistanceSensor (он расположен спереди под острым углом к горизонту), с помощью которого робот замечает яму спереди. А также мы добавили ColorSensor в качестве подсветки для лучше работы камеры (рис.3).

Програмное обеспечение

Общее устройство программного обеспечения

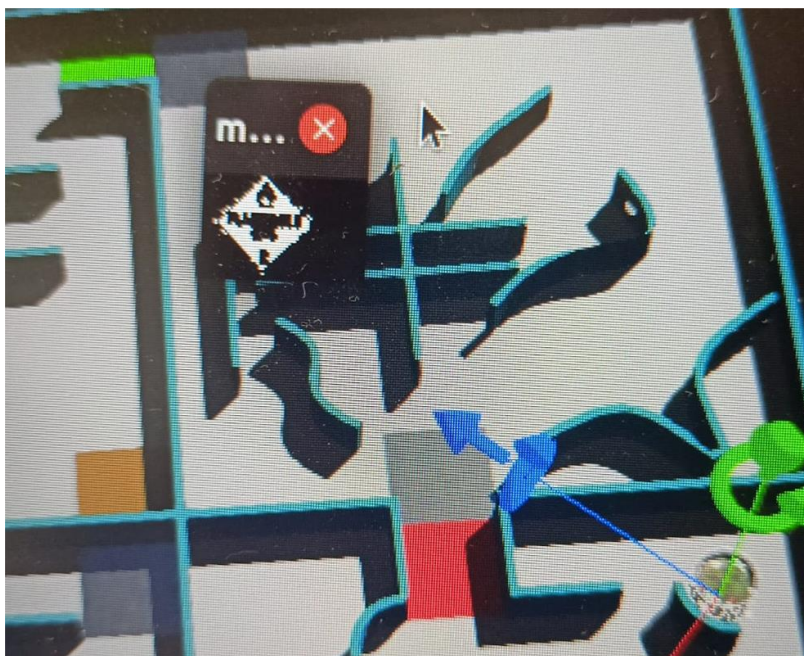
- Для написания алгоритма мы выбрали язык программирования C++, потому что он выигрывает у Python в производительности и скорости. До функции main мы подключаем библиотеки, в том числе от Webots и OpenCV, инициализируем команды, объявляем основные константы. В main инициализируем необходимые датчики и моторы и привязываем их работу к шагу времени. В бесконечном цикле while (robot->step(timeStep) != -1 вначале идёт проверка переехал ли робот в новую зону, если переехал, то вносятся начальные показания для некоторых переменных, далее идёт проверка на отсутствие прогресса, если такое обнаруживается, то робот проезжает вперёд - назад для проверки своей ориентации в пространстве посредством компаса и гироскопа. После идёт блок, предназначенный для передвижения робота и напоследок, проверка на антизалипание, что бы если робот застрял, он отъезжал назад...

Навигация

- Алгоритм навигации робота построен на карте, изначально, робот высчитывает координаты grs передней и левой зоны, отдавая предпочтения поворота на неисследованную территорию, если же все доступные пути уже исследованы робот движется по алгоритму действующей руки.

Распознавание “меток на стенах”

- С помощью камеры Erebus и библиотеки OpenCV 4 происходит обнаружение и распознавание “знаков опасности” и “жертв”. Жертвы и Hazmat signs мы обнаруживаем с помощью датчика камеры и библиотеки OpenCV 4. Для начала настраивается бинаризация для 1 из 3 цветов: красного, жёлтого и чёрного. После бинаризации определяем контуры и проверяем самые большие контуры, подходят ли они. Далее если нашлись подходящие контуры для жёлтой бинаризации, то мы определили Hazmat sign- Organic Peroxide, если же не найдены подходящие жёлтые, но есть подходящие красные, то это Flammable Gas, если же обнаружены чёрные контуры, подходящего размера, то вычисляем 3 точки прямоугольника этого контура: верхнюю срединную, среднюю срединную и нижнюю срединную, с помощью них мы и определяем, какая это буква, и ищем в них контур на чёрной бинаризации: если верхний, средний и нижний - чёрные, то это s, если нижний и верхний - не чёрные, а средний - чёрный, то это H, а если нижний - чёрный, а средний и верхний - нет, то это U;



Распознавание метки на стене путём бинаризации. Рис.4

Оценка конкурентоспособности

- На данный момент, разработанный нами робот может безопасно перемещаться по карте распознавая все «метки на стенах».
- В будущем планируется разработать алгоритм создания карты

Заключение

- Наша команда изучила принципы работы некоторых методов и функций OpenCV, узнала и научилась применять возможности некоторых датчиков.

Благодарности

Наша команда благодарит нашего тренера - Косаченко Сергея Викторовича, ОГБОУ «Томский физико-технический лицей» и организаторов соревнований!

Ссылки

- <https://cyberbotics.com/doc/reference/camera?tab-language=c++> – Webots Documentation;
- <https://www.youtube.com/@KajalGada> – Kajal Gada;
- <https://www.youtube.com/@StormingRobots> – Storming Robots.

Наш Github

- <https://github.com/AngelinaBan/RoboCup-Rescue-Maze->