

Интеллектуальные робототехнические системы

2022/23 учебный год

Инженерный тур

Общая информация

Необходимо выполнить автоматизацию процесса погрузки и разгрузки на складе (открытой площадке). Известно, что на две площадки доставили груз, группа из роботов должна забрать весь этот груз и распределить его по складу так, чтобы управляться за минимальное время. Необходимо учитывать, что через определенное время груз со склада нужно отвезти обратно в зону погрузки, чтобы транспорт мог доставить его в пункт назначения.

Легенда задачи

Изменение приоритетов во внешней торговле привело к существенному изменению логистики товарных потоков. Склад малой мощности оказался востребован как хаб хранения и распределения большого объема потребительских товаров, для чего необходимо повысить производительность. Одно из решений такой задачи — автоматизация процессов ПРР (погрузочно-разгрузочных работ).

Требования к команде и компетенциям участников

Количество участников в команде: 3–4.

Компетенции, которыми должны обладать члены команды:

(Роли, которые должны быть представлены в команде)

1. Математик-алгоритмист

Подбор и разработка алгоритмов для программистов: управления и навигации, создания графа занятости, алгоритмов поиска пути в графе, построения траектории и т. д.

2. Программист системы управления и навигации робота

Программирование системы управления роботом, планирования пути и следование по нему. Понимание как работает теория графов и как настраиваются регуляторы.

3. Программист машинного зрения

Программирование системы машинного зрения, определение координат объектов на изображении, классификации объектов, определения препятствий

4. Капитан команды

Распределение задач, расстановка приоритетов, распределение ресурсов, разрешение споров, проектирование архитектуры решения, программирование высокоуровневой интеллектуальной системы управления, создание связи между системой управления робота, системой навигации и планирования движения и системой машинного зрения

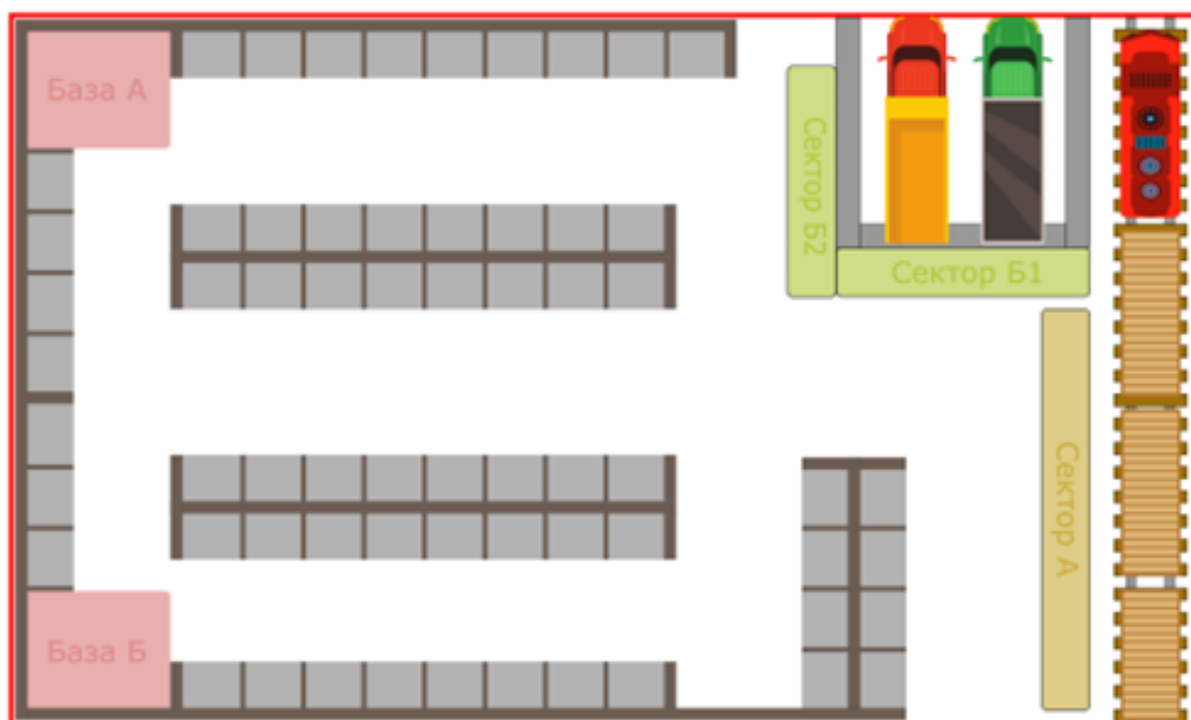
Оборудование и программное обеспечение

Наименование	Описание
Ноутбук	Для программирования роботов и решения задачи
Операционная система Linux: https://ubuntu.com	Для программирования роботов и решения задачи, поддерживает ПО для роботов
Робоплатформа РОББО	Для транспортировки груза
Набор расширения РОББО	Датчики к робоплатформе РОББО и захват (гриппер) для захвата и транспортировки груза
Камера	Для захвата изображения над полем и слежения за роботом
Поле 3 × 5 м	Поле с отметками для имитации склада (зона погрузки/разгрузки, складские ячейки, свободное пространство для передвижения робота и др.)

Описание задачи

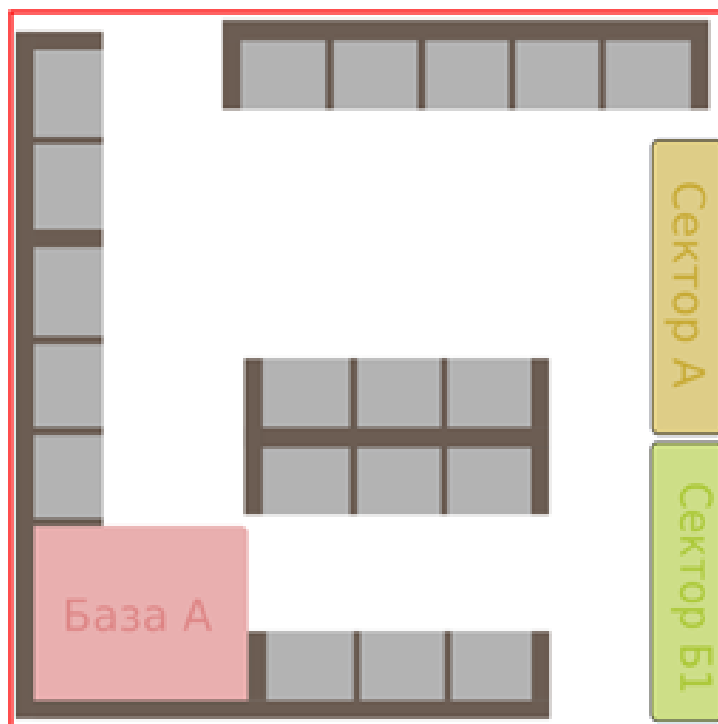
Основное поле проведения соревнований

Поле высотой 3 метра и шириной 5 метров представляет схему площадки с шестью зонами хранения, тремя зонами погрузки/разгрузки (Сектор Б1, Б2, А). Секторы Б1 и Б2 предназначены для работы с фурами, Сектор А для поездов. База А и Б может использоваться для старта роботов в задаче погрузки и разгрузки. Поле рассчитано на работу 3–5 роботов в один промежуток времени, проведения тестов алгоритмов навигации, компьютерного зрения и управления, а также для проведения раундов с оцениванием. Для проведения тестов участникам выделяются слоты времени, на которые капитаны должны записывать каждое утро в течение всех дней олимпиады.

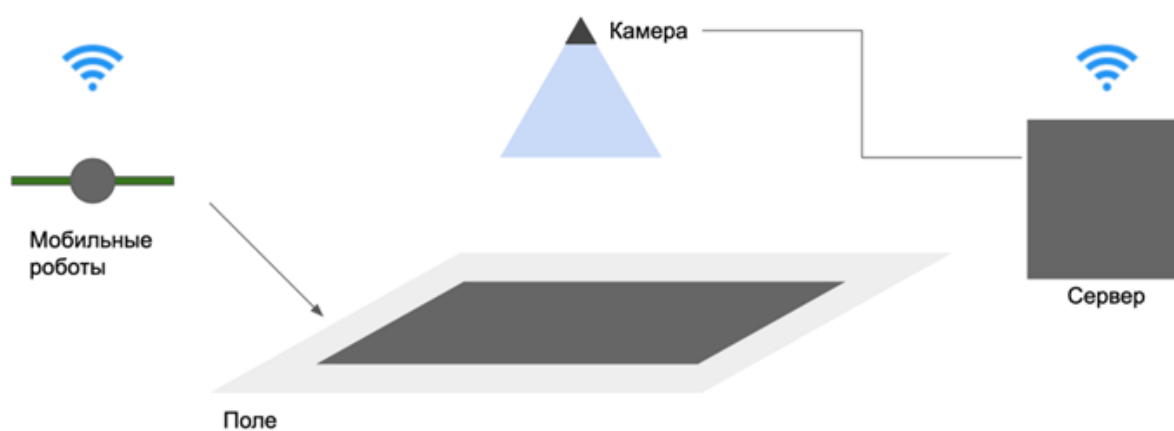


Тестовое поле проведения соревнований

Поле 2×2 метра, присутствуют все те же зоны, что и на основном, размеры ячеек хранения идентичны. На рисунке ниже можно увидеть 4 зоны хранения, две зоны погрузки/разгрузки (Сектора А и Б1) и Базу А для старта движения роботов. Одновременно площадка вмещает в себя 2–3 робота. Тестовое поле выделяется по одной штуке на 3 команды. Предназначено для непрерывного тестирования алгоритмов навигации, компьютерного зрения и управления.



Оснащение основного поля



Компьютерное зрение

1. Определение ячейки хранения товара

Команде называются три случайных номера МТР, расположенных на поле. Капитан команды последовательно вводит три озвученных номера и нажимает «Ввод». Программа должна обвести ячейки на изображении поля, в которых находятся МТР в рамки синего, зеленого и красного цвета. Синяя рамка ячейки соответствует максимальному из названных номеров МТР, красная рамка — минимальному из названных номеров МТР. За каждую верно выделенную ячейку команда получает **1 балл**. **Максимум 3 балла**.

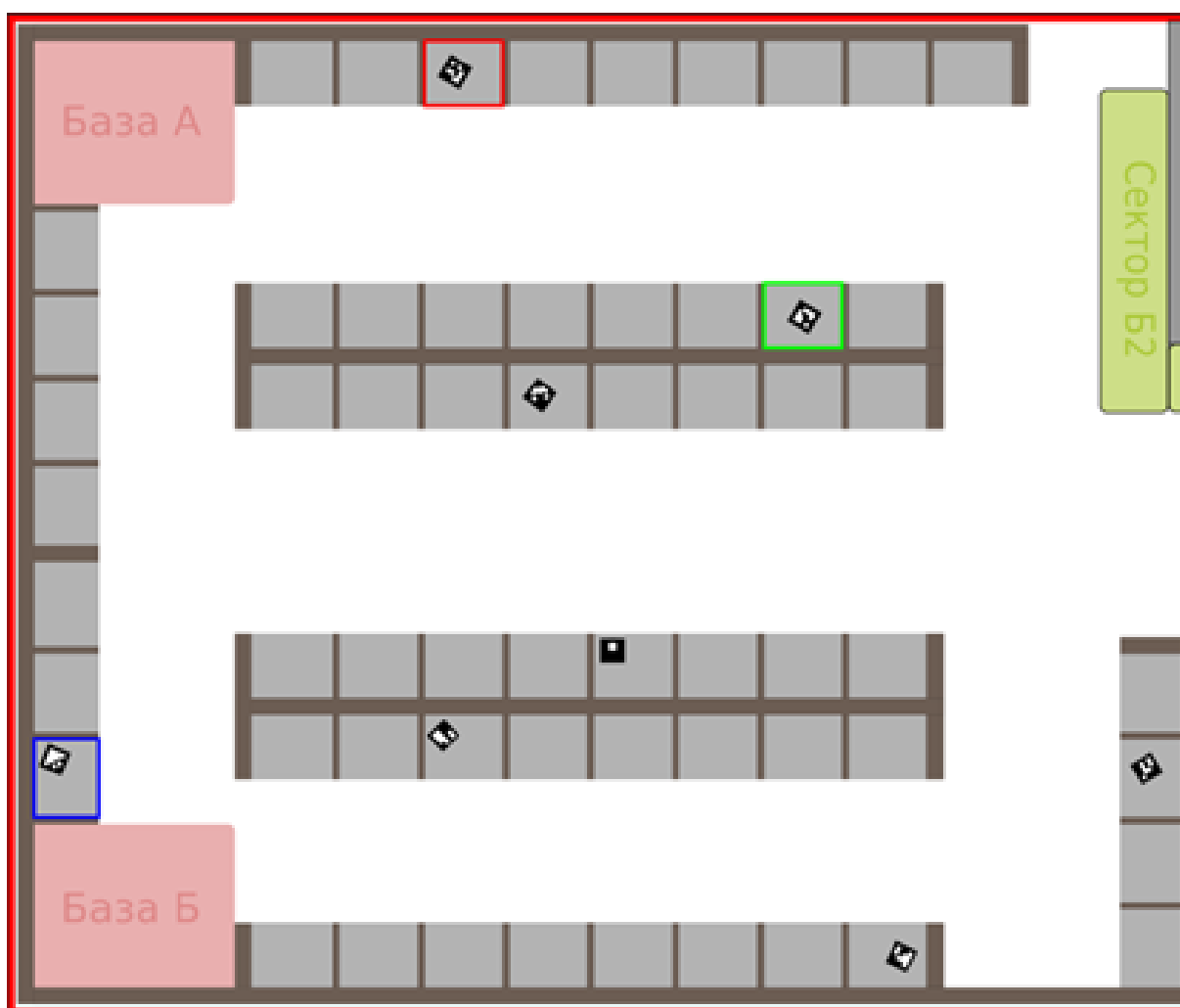


Рис. VI.2.1. Пример верно выполненного задания (3 балла) для следующих входных данных: 5 4 8

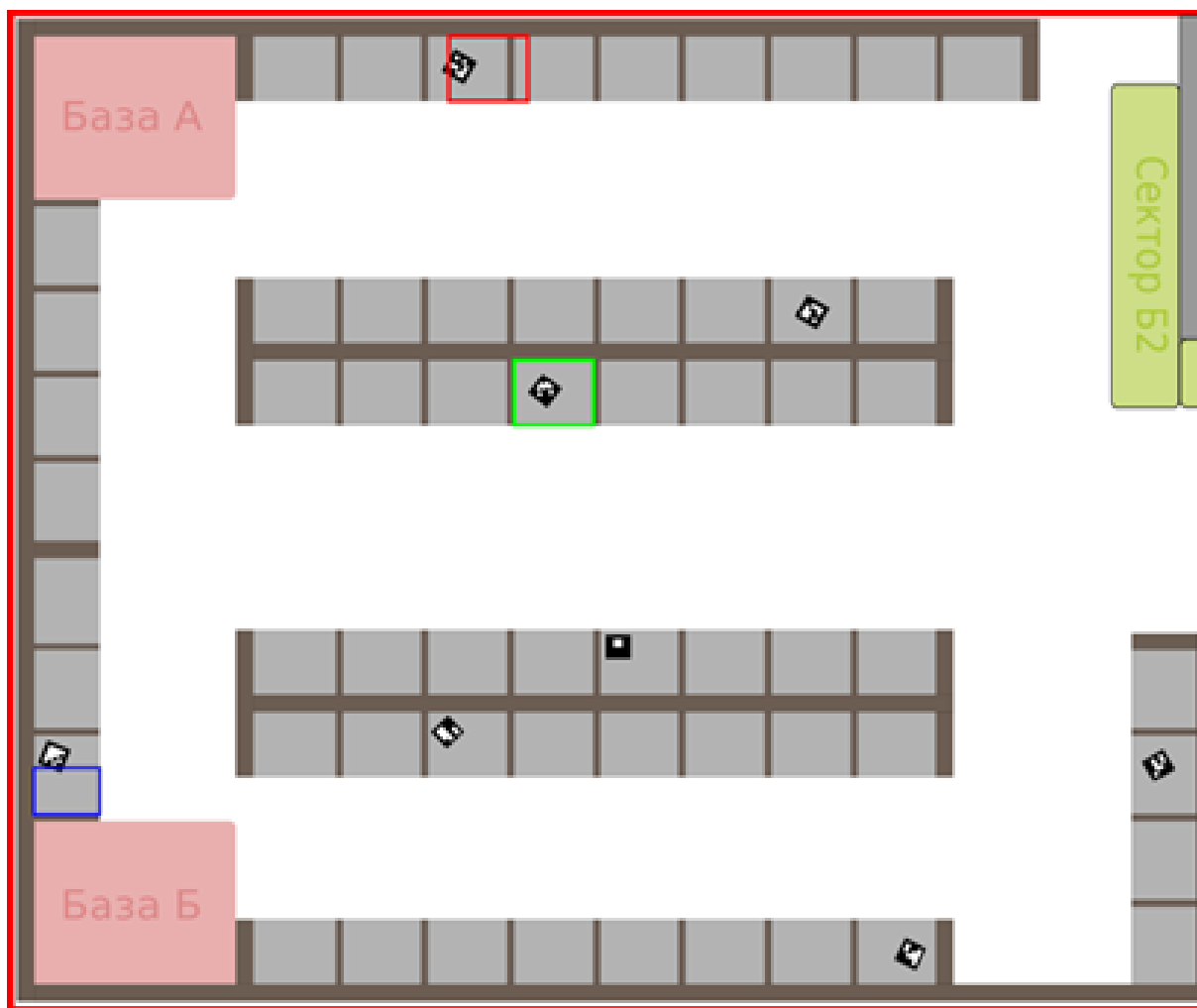


Рис. VI.2.2. Пример неверно выполненного задания (0 баллов) для следующих входных данных: 5 4 8

2. Определение координат роботов

На поле в случайных местах устанавливаются 3 робота с произвольной ориентацией. Необходимо определить положения роботов на карте в пиксельной системе координат. При нажатии клавиши «Ввод» капитаном команды, программа должна показать изображения с камеры, где красной линией выделены края маркера робота и рядом написан его *id*. Линия должна четко проходить по краю, если она явно смещена, ответ не засчитывается. За каждое верно определенный маркер (линия + *id*) начисляется **1 балл. Максимум 3 балла.**

Пример выходных данных программы

```
123 548
376 984
782 451
```

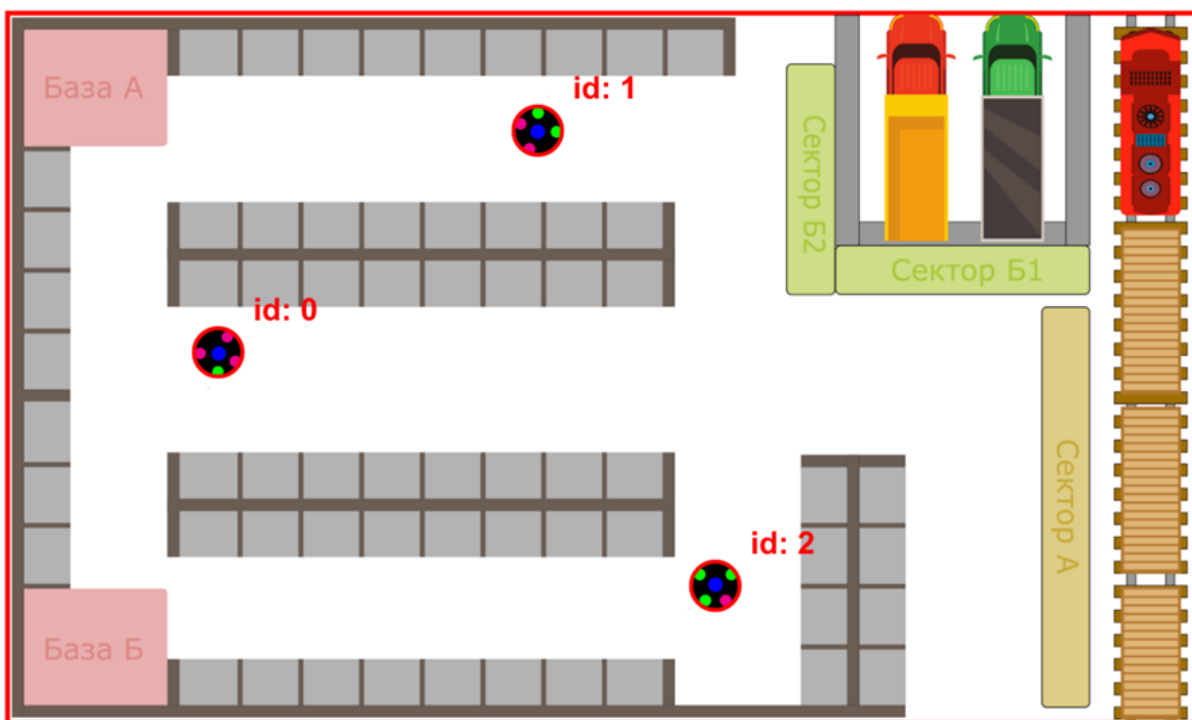


Рис. VI.2.3. Пример правильного ответа

3. Построение карты занятости ячеек склада (индикация занятых ячеек)

Ячейки склада случайным образом заполняются МТР. Команде необходимо продемонстрировать работоспособность программы, которая, основываясь на изображении полученной с камеры определит какие ячейки заняты, а какие свободны. При нажатии клавиши «Ввод» программа должна выделить зеленым цветом незанятые ячейки и красным цветом занятые. При отсутствии ошибок команда получает **3 балла**. За каждую ошибку у команды снимается **1 балл**. Минимальное количество баллов: 0 баллов.

Управление роботом

1. Визуальное управление

Команда должна написать программу для управления роботом на основе обратной связи по изображению.

При нажатии клавиши «Ввод» программа должна выполнять следующие действия:

1. Открыть окно с видеотрансляцией поля и пометить положение робота на поле.
2. При нажатии судьей курсором мыши по изображению, программа должна, используя удаленное управление роботом посредством bluetooth, заставить робота стабилизироваться в заданной точке. Точка выбирается в соответствии с требованием об отсутствии возможных препятствий на пути и при условии ее достижимости.

Судья последовательно, ожидая достижения целей и полной остановки робота, задает 3 цели для достижения роботом. За каждую достигнутую роботом цель команда получает **1 балл. Максимум 3 балла.**

2. Захват объекта; выполнение простых команд с объектом

Команда должна написать программу для захвата МТР из зоны хранения. Робот устанавливается за 10 см до предполагаемой ячейки хранения с МТР.

При нажатии клавиши «Ввод» в программе робот, управляемый посредством bluetooth должен выполнить следующие действия.

1. Подъехать к ячейке с МТР.
2. Захватить МТР.
3. Отъехать к стартовому положению.

За каждое совершенное действие команда получает по **1 баллу. Максимум 3 балла.**

Навигация

1. Позиционирование робота в заданной координате

Робот устанавливается в случайной свободной (белой) клетке поля.

При нажатии клавиши «Ввод» программа должна выполнять следующие действия:

1. Открыть окно с видеотрансляцией поля и пометить положение робота на поле аналогичным образом, как в Задаче «**Определение координат роботов**».
2. При нажатии судьей курсором мыши по изображению, программа должна выделить красным цветом цель, спланировать движение и, используя удаленное управление роботом, посредством bluetooth, заставить робота следовать по маршруту с заданной точке. Центр робота должен оказаться в центре целевой точки. Точка выбирается в соответствии с требованием ее достижимости.

Судья последовательно, ожидая достижения целей и полной остановки робота, задает 3 цели для достижения роботом. За каждую достигнутую роботом цель команда получает **1 балл.**

2. Планирование безопасного движения группы роботов (2 робота, без столкновений)

Команда должна разработать программу, которая, основываясь на показаниях камеры позволит безопасно управлять группой роботов (2 робота). Роботы устанавливаются в случайных свободных (белых) клетках поля. При нажатии клавиши «Ввод» программа должна выполнять следующие действия:

1. Открыть окно с видеотрансляцией поля и пометить положение роботов на поле аналогичным образом, как в задаче «**Определение координат роботов**».
2. При нажатии судьей курсором мыши по изображению, программа должна запомнить и выделить первую цель. При повторном нажатии в другом месте программа должна выделить вторую цель.

- Используя удаленное управление роботом посредством bluetooth, имея информацию о карте и положении роботов спланировать движение роботов к точке с избеганием столкновений с другими роботами.
- Роботы, под управлением программы должны выполнить следование по найденному маршруту, избегая столкновений с препятствием и друг с другом. Робот, с младшим id маркера, должен приехать в первую заданную судьей точку, а робот со старшим id маркера во вторую. Точки выбираются в соответствии с требованием ее достижимости.

При отсутствии ошибок команда получает **3 балла**. За каждую ошибку, столкновение или не достижение цели у команды снимается **1 балл**. Минимальное количество баллов: 0 баллов.

Погрузка и разгрузка (Задача инженерного тура)

Участникам необходимо реализовать программу для управления группой из 3–5 роботов, которые выполняют ПРР из зоны хранения в зону погрузки/разгрузки (Сектор А или Б) и наоборот.

Перед началом задания проводится инициализация системы:

- Роботы размещаются на поле с произвольным положением и ориентацией
- В ячейки зоны хранения устанавливаются МТР в соответствии со списком для раунда
- В свободной для передвижения зоне склада устанавливаются статичные препятствия
- Чтением сообщений системы по протоколу TCP/IP выдается лист заказов (в виде строки) на ПРР для команды. Лист содержит информацию о МТР и о зоне доставки. Здесь
 - `task_id` — идентификатор задачи,
 - `cargo_id` — идентификатор МТР, который нужно перевезти,
 - `sector` — значение сектора погрузки/разгрузки (есть сектор А и Б)
 - `action` — тип действия, отвезти в зону погрузки/разгрузки `to` и из нее `from`.

Пример:

```
[
    {'task_id': 101, 'cargo_id': 3, 'sector': 'A', 'action': 'to'},
    {'task_id': 102, 'cargo_id': 4, 'sector': 'A', 'action': 'to'},
    {'task_id': 103, 'cargo_id': 4, 'sector': 'B', 'action': 'to'},
    {'task_id': 104, 'cargo_id': 9, 'sector': 'A', 'action': 'from'},
    {'task_id': 105, 'cargo_id': 5, 'sector': 'B', 'action': 'to'},
    {'task_id': 106, 'cargo_id': 6, 'sector': 'B', 'action': 'to'},
    {'task_id': 107, 'cargo_id': 1, 'sector': 'A', 'action': 'from'},
    {'task_id': 108, 'cargo_id': 3, 'sector': 'B', 'action': 'to'}
]
```

- На подготовку к старту команде дается 5 минут.

Для успешного выполнения заказа робот должен:

- Отправить в систему сообщение (по протоколу TCP/IP) со строкой о начале выполнении задания, например, «START 2131», где после слова START указывается id задания.

2. Отправиться к ячейке с МТР с заданным `cargo_id`, избегая столкновений с объектами.

3. Сразу как участник подъехал к объекту и намерен начать захват он должен отправить в систему сообщение (по протоколу TCP/IP) со словарем, например:

```
{
    task_id: 2131,
    action: "reached"
}
```

4. Захватить объект с заданным `cargo_id` и отправить в систему сообщение (по протоколу TCP/IP) со словарем, например:

```
{
    task_id: 2131,
    action: "grabbed"
}
```

5. Доставить объект в зону назначения (это может быть как любая зона хранения на выбор участника в случае `'action': 'from'`, так и заданный сектор А или Б в случае `'action': 'to'`). Аналогично отправить в систему сообщение (по протоколу TCP/IP) со словарем, например:

```
{
    task_id: 2131,
    action: "delivered"
}
```

6. Положить объект в корректную зону и отправить в систему сообщение (по протоколу TCP/IP) со словарем, например:

```
{
    task_id: 2131,
    action: "put"
}
```

7. После того, как робот выполнил задание, участник должен отправить сообщение, например, «END 2131», где после слова END указывается id задания. Важно соблюдать порядок выполнения заданий. Нельзя пропускать какое-то задание, то есть, например, нельзя начать выполнение задания 103, если не начато Задание02. Также, нельзя начать выполнять заданий больше, чем имеется роботов. То есть, например, если есть 3 робота, нельзя сообщить системе, что выполняется больше 3 заданий.

Для общения с системой по протоколу TCP/IP на языке Python, рекомендуется использовать следующие команды:

- `recv()` — для чтения сообщений системы (для получения задания);
- `sendall()` — для отправки сообщений системе, важно, что данный метод принимает только объекты в битовом представлении.

Пример программы для общения с системой

```
import socket
import sys
import json
import time
# Create a TCP/IP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```

# Connect the socket to the port where the server is listening
server_address = ('localhost', 10000)
# Messages examples
msgs = ["START 101", "START 102", {"task_id":102, "action":"reached"},
↪ {"task_id":102, "action":"reached"}, {"task_id":102, "action":"delivered"},
↪ {"task_id":102, "action":"put"}, "END 102"]
sock.connect(server_address)

try:
    # get message from server
    data = (sock.recv(1024)).decode()
    # print task from refbox
    print('received', data)
    time.sleep(1)
    for message in msgs:
        # Send data
        print('sending', message)
        if isinstance(message, str):
            # if string
            sock.sendall(message.encode())
        else:
            # if dict
            user_encode_data = json.dumps(message, indent=2).encode()
            sock.sendall(user_encode_data)

        time.sleep(1)

finally:
    print('closing socket')
    sock.close()

```

Ссылка на репозиторий с системой: https://gitlab.com/StriderOne/nto_refbox.git.

Участник получает баллы за:

- приехать вплотную к целевому МТР — 1 балл;
- осуществить захват целевого МТР — 1 балл;
- отвезти МТР в пункт назначения — 1 балл;
- освободить МТР в зоне назначения — 1 балл.

Участник теряет баллы за:

- легкое столкновение с препятствие — 1 балл;
- наезда на линию разметки границы поля — 1 балл.

Участник получает 0 баллов и задача перезапускается:

- сильное столкновение (предмет сдвинулся);
- выезд за пределы поля;
- наезд на линии разметки стен.

Результат: по итогу выполнения финального задания участники должны написать систему из двух программ (для робота и серверного ПК), способную:

1. С помощью методов компьютерного зрения определять основные объекты на

-
- складе: робота, МТР, препятствия, стены и ячейки зоны хранения.
2. Строить карту по результатам пункта 1.
 3. Строить путь по карте из пункта 2 для нескольких роботов.
 4. Находить кратчайший путь для нескольких роботов.
 5. Управлять роботом по заданной траектории.
 6. Управлять роботом для захвата и освобождения МТР.
 7. Выполнять коммуникацию между роботами и серверным ПК.

Система оценивания

Баллы за прохождение «Задание. Компьютерное зрение»

Максимальное количество баллов за тест — 9.

1. Определение места хранения товара. В случае, когда Маркер находится не над заданным МТР балл снимается. Так, снимается до 3 баллов.
2. Определение координаты робота. В случае, если Маркер находится за пределами робота балл снимается. Так, снимается до 3 баллов.
3. Построение карты занятости ячеек склада. Если маркер занятых ячеек находится за пределами ячейки, снимается по 1 баллу за 1 ошибку. Так, снимается до 3 баллов.

Баллы за прохождение «Задание. Управление роботом»

Максимальное количество баллов за тест — 6.

1. Визуальное управление. В случае, если робот не достиг цели или не совершил полную остановку по достижении цели, балл снимается. Так, снимается до 3 баллов
2. Захват объекта;выполнение простых команд с объектом. В случае, если робот не совершил одно из трех заданных действий балл снимается. Так, снимается до 3 баллов

Баллы за прохождение «Задание. Навигация»

Максимальное количество баллов за тест — 6.

1. Позиционирование робота в заданной координате. В случае, если робот метка будет за пределами робота, тогда балл отнимается. Так снимается до 3 баллов
2. Планирование безопасного движения группы роботов (2 робота, без столкновений). При отсутствии ошибок команда получает 3 балла. За каждое столкновение или не достижение цели у команды снимается 1 балл. Так, снимается до 3 баллов.

Баллы за прохождение задания инженерного тура

1. Участник получает баллы за:

-
- приехать вплотную к целевому МТР — 1 балл;
 - осуществить захват целевого МТР — 1 балл;
 - отвезти МТР в пункт назначения — 1 балл;
 - освободить МТР в зоне назначения — 1 балл.
2. Участник теряет баллы за:
- легкое столкновение с препятствие — 1 балл;
 - наезда на линию разметки границы поля — 1 балл.
3. Участник получает 0 баллов, и задача перезапускается:
- сильное столкновение (предмет сдвинулся);
 - выезд за пределы поля;
 - наезд на линии разметки стен.
4. Баллы за все попытки выполнения задачи инженерного тура суммируются
- Командный балл рассчитывается как

Баллы за тесты + Сумма баллов за все попытки пройти задачу инженерного тура.

Например, участник сдал три теста, каждый по 6 баллов и 2 раза сдал задачу финала на 10 баллов. Итого мы получаем $6 + 10 + 10 = 26$ баллов.

Далее результат приводится к 100-балльной системе. Для этого вычисляется коэффициент

$$k = \frac{100}{\text{макс. балл команды}},$$

далее баллы всех команд умножаются на этот коэффициент.

Итоговый балл участника равен:

$$0,15 \cdot (\text{Математика}) + 0,15 \cdot (\text{Информатика}) + 0,7 \cdot (\text{Командный балл}).$$

В случае равенства баллов участников и невозможности определить призеров и победителей, то их программный код оценивается и выставляется оценка от 0 до 10. Команды также могут попросить рассказать о своем решении, выступление будут оценивать также от 0 до 10.

Решение задачи

Компьютерное зрение

1. Определение ячейки хранения товара

Для успешного решения задачи можно было воспользоваться стандартной библиотекой `cv2.aruco`, которая позволяет определить номера всех маркеров на изображении, а также положение их центров в пикселях. Далее, нужно было определить границы всех ячеек на изображении, для этого можно было воспользоваться различными стандартными операциями морфологии, трешхолда, так как цвет ячеек все время оставался неизменным.

2. Определение координат роботов

Решение данной задачи схоже с решением прошлой. Пользуясь операциями над изображениями и зная цвета кругов на своем роботе, можно определить координаты пикселей центров кругов, после этого несложно определить координаты пикселей центра самого робота, а также его ориентацию, так как цветные круги однозначно определяют ориентацию робота.

3. Построение карты занятости ячеек склада (индикация занятых ячеек)

Чтобы решить данную задачу, можно было воспользоваться решением первой задачи, отличие в том, что нужно пометить все ячейки, и если в пределах её границ есть центр какого-либо маркера, то пометить границы данной ячейки красным цветом, а иначе — зеленым.

Управление роботом

1. Визуальное управление

Решение данной задачи строится на решении задачи определения координаты робота. Получив координаты центра робота в пикселях, и зная координаты точки, указанной судьей, можно построить регулятор по ошибке по расстоянию и по углу отклонения курса робота, от заданной точки. Например, можно было выбрать пропорциональный регулятор. По условию задачи, на пути отсутствуют препятствия и точка достижима, это гарантирует то, что такой регулятор сойдется (при условии правильно подобранных коэффициентов).

2. Захват объекта; выполнение простых команд с объектом

Для решения данной задачи необходимо воспользоваться ультразвуковым датчиком, который был доступен каждой команде. Решение можно разбить на несколько этапов:

1. Поиск объекта: предполагается, что вокруг робота находится только один объект, значит роботу нужно вращаться вокруг своей оси, до тех пор, пока значения, полученные с ультразвукового датчика, не станут близки к 10 см (именно такое расстояние до объекта).
2. После того, как робот нашел объект, ему необходимо подъехать к нему поближе. Для того чтобы сделать, рекомендуется также воспользоваться регулятором, но теперь только по ошибке по расстоянию. Такой подход позволит роботу остановиться на нужном расстоянии от объекта с большой точностью.
3. Далее, роботу нужно захватить объект, заранее посчитав, на какую величину надо сжимать захват. После этого, роботу необходимо отъехать назад к стартовому положению, сделать это можно было, используя встроенные в колеса робота энкодеры, и запомнив расстояние, которое прошел робот до объекта.

Навигация

1. Позиционирование робота в заданной координате

Решение данной задачи строится на решении задания «Визуальное управление». Отличие состоит в том, что теперь на пути робота от стартовой позиции до конечной могут быть препятствия. Для решения этой задачи можно было воспользоваться стандартными алгоритмами навигации. Один из алгоритмов заключается в построении графа в рабочей области, для этого можно было воспользоваться алгоритмами RRT (https://en.wikipedia.org/wiki/Rapidly-exploring_random_tree) или PRM (https://en.wikipedia.org/wiki/Probabilistic_roadmap). После построения графа, поняв, в какой вершине находится робот, а в какой вершине находится конечная позиция, задача сводится к нахождению кратчайшего пути в графе. Для решения можно было воспользоваться алгоритмом Дейкстры (https://ru.wikipedia.org/wiki/Алгоритм_Дейкстры). Получив кратчайший путь, роботу надо двигаться последовательно по вершинам, движение между вершинами можно было осуществить с помощью решения задания «Визуальное управление».

2. Планирование безопасного движения группы роботов

Решение данной задачи строится на решении задания «Позиционирование робота в заданной координате». При таком подходе роботы двигаются только вдоль вершин графа. Тогда, зная пути роботов, нужно проверить две ситуации:

1. Роботы в один тот же момент времени могут прийти в одну и ту же вершину
2. В один момент времени при прохождении своих отрезков траектории могут пересечься, проверять это можно, например, проверяя на пересечение отрезки, соединяющие вершины графа

При выявлении любой из двух ситуаций необходимо затормозить одного из роботов, тем самым давая другому роботу пройти участок пути без столкновений.

Погрузка и разгрузка (Задача инженерного тура)

Для решения данной задачи нужно было воспользоваться решениями всех прошлых задач. Решение разделяется на несколько этапов:

1. Построение графа в рабочей области, как в задании 3.
2. Детектирование грузов и определение их id.
3. Получив заказ от тестирующей системы, необходимо грамотно спланировать траектории роботов. Например, можно было продумать движение роботов так, чтобы после выполнения очередного задания, робот приступал к заданию, для которого груз находится ближе всего к данному роботу, это позволит сэкономить время при выполнении задания. При разгрузке, разумнее ставить грузы поближе к зоне разгрузки, с целью экономии времени.
4. Распределив задания между роботами, можно приступать к реализации заданий. Здесь задание также сводить к заданию 3. Зная информацию о каждом задании, не трудно определить начальную и конечные точки для каждого робота.
5. При приближении робота к объекту необходимо воспользоваться решением за-

дачи с захватом объекта.

6. После захвата объекта, аналогично пункту 4 нужно спланировать движение робота с грузом до конечной точки, по достижении точки, груз надо отпустить.
7. Далее необходимо вернуться к пункту 4.

Материалы для подготовки

Математика:

- Суть матанализа: <https://youtu.be/qd0rzmSGPWg>.
- Сущность линейной алгебры. Введение: <https://youtu.be/RNTRYicPvWQ>.

Теория управления и навигация:

- Хардкорная математика для робототехников: <https://www.lektorium.tv/hardcoremath-materials>.
- Control of Mobile Robots-1.1 Control of Mobile Robots: <https://youtu.be/aSwCMK96N0w>.

Машинное зрение:

- OpenCV-Python Tutorials: https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html.
- Курс. Компьютерное зрение. Базовый уровень: <https://avt.global/cv>.

Навигация и планирование пути:

- Казаков К. А. и др. «Обзор современных методов планирования движения».
- Чему робототехника может научить игровой ИИ: <https://habr.com/ru/post/349044/>.
- Алгоритм планирования пути PRM: <https://russianblogs.com/article/8569373974/>.
- PythonRobotics: <https://github.com/AtsushiSakai/PythonRobotics#path-planning>.