

1. ФИНАЛЬНЫЙ ЭТАП

Задача командного тура

1.1. Легенда

Недалёкое будущее, в автоматических логистических центрах практически нет людей, всю работу выполняют роботы-погрузчики, которыми управляет интеллектуальное ПО по распределению задач.

Несмотря на отсутствие человеческого фактора, не следует думать, что в таких центрах никогда не будет проблем. Форс-мажор может произойти в любой момент и система из роботов и ПО должна уметь справляться с такими ситуациями.

Поэтому для финальной задачи профиля «Интеллектуальные робототехнические системы» предлагается рассмотреть следующий эпизод: в логистическом центре произошла перезагрузка всех систем, что привело к сбросу информации о местоположениях работающих в данный момент роботов-погрузчиков, а также карты данной части логистического центра.

В начале выполнения задания считается, что робототехническое устройство активируется в какой-то секторе логистического центра. Один из роботов выполняет считанные с `arTag` маркера команды, а затем ожидает прибытия второго робота в смежный сектор. Первый робот должен найти второго, доехав до смежного с ним сектора, а после вернуться в один из секторов старта. Структура логистического центра не известна заранее. При перемещении роботы не должны сталкиваться, а также повреждать логистический центр.

Задача участников Олимпиады — разработать программу управления робототехнического устройства для выполнения задания описанного выше.

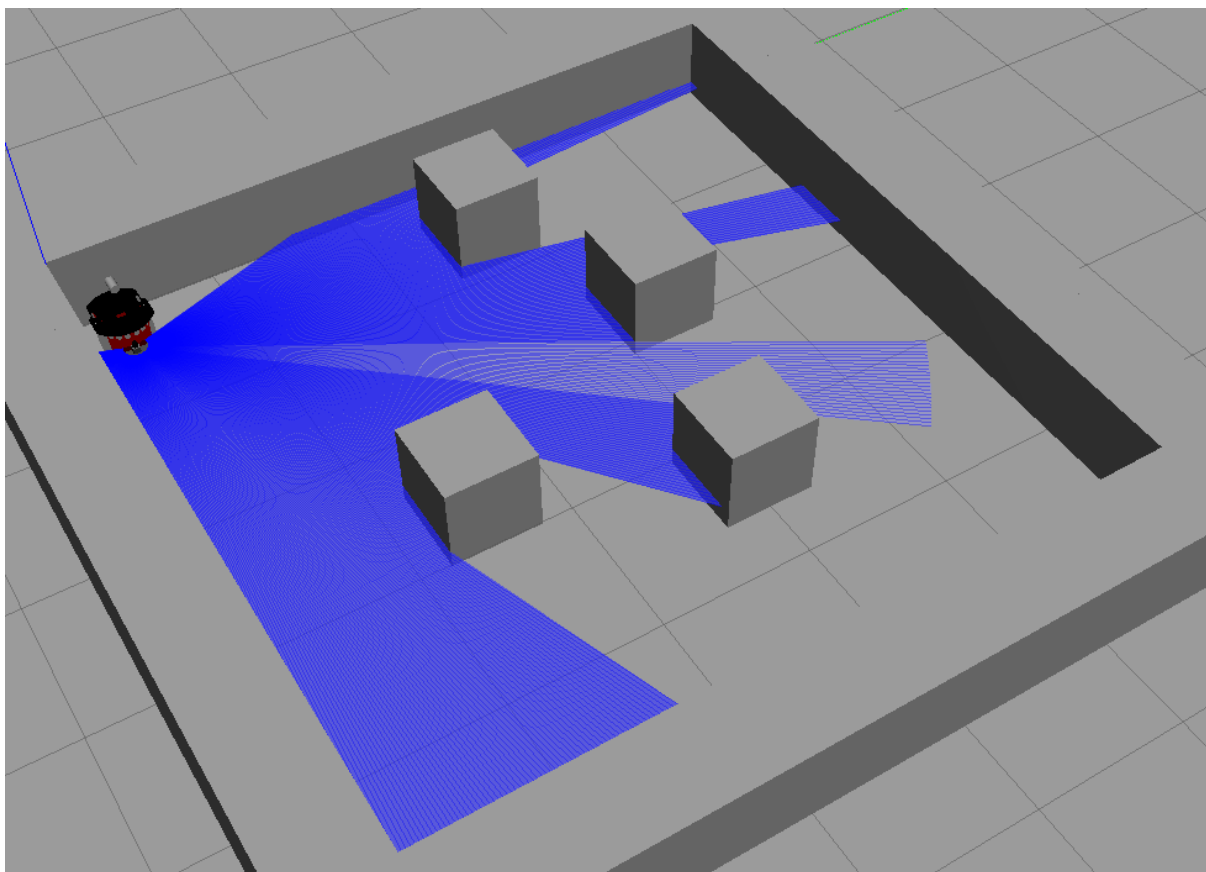


Рис. 1.1: Полигон для запуска робототехнических устройств на финале Олимпиады НТИ

1.2. Набор заданий

Решение командной задачи разбито на три этапа. Первые два этапа итеративно подводят участников к решению полной финальной задачи, осуществляемому во время последнего третьего этапа. На каждом этапе в проверку решения заданий данного этапа входят:

- способность проверить гипотезу о работоспособности алгоритма через демонстрацию решения в симуляторе;
- полнота решения задания конкретного этапа;
- воспроизводимость результатов — робототехническое устройство участников должно неоднократно выполнить требуемые действия.

Первый этап

Задача: Робот начинает свое движение в углу полигона. На полигоне имеются четыре тумбы различных цветов. Цвет тумбы, наименее удаленной от стены полигона, задаёт направления координат и соответственно начальные координаты. Тумбы располагаются параллельно стенкам и координаты их центра задаются следующим образом: $(0, 5i; 0, 5j)$, где i, j — натуральные числа (т.е. тумбы могут быть смещены на пол сектора).

Включая содержательные задачи:

- Построение карт местности робототехническими системами.
- Реализация алгоритмов вычисления усредненного цвета графического поля с помощью камеры.
- Реализация алгоритмов определения координат положения.

Второй этап

Задача: робот начинает своё движение возле желтой тумбы с случайной стороны и с неизвестной ориентацией. Необходимо найти ARTag маркер и расшифровать координаты, закодированные на нем.

Включая содержательные задачи:

- Реализация поиска ключевых объектов на изображении с камеры.
- Реализация алгоритмов перемещения до ключевого кадра на камере.
- Реализация алгоритмов исключения лишних элементов из кадра.
- Реализация алгоритмов компьютерного зрения: считывание ARTag меток без использования дополнительных библиотек.
- Декодирование бинарного кода с использованием кода Хэмминга.

Третий этап

Задача: Робот начинает свое движение в углу полигона. На полигоне имеются четыре тумбы различных цветов. Цвет тумбы наименее удаленной от стены полигона задаёт направления координат и соответственно начальные координаты. На желтой тумбе с неизвестной стороны располагается ARTag метка, кодирующая координаты финиша. Тумбы располагаются параллельно стенкам и координаты их центра задаются следующим образом: $(0, 5i; 0, 5j)$, где i, j — натуральные числа (т.е. тумбы могут быть смещены на пол сектора).

Включая содержательные задачи:

- Калибровка камеры робототехнического устройства;
- Реализация алгоритмов составления карты неизвестной местности и локализация на данной местности;
- Реализация алгоритмов перемещения на пустой местности без опорных сооружений;
- Реализация алгоритмов компьютерного зрения: считывание ARTag меток без использования дополнительных библиотек;
- Декодирование бинарного кода, с использованием кода Хэмминга;

1.3. Описание модели логистического центра

Полигон - квадратное поле 4800×4800 мм., разделенное на квадратные сектора 600×600 мм. Некоторые места полигона недоступны для посещения робототехническим устройством и представляют из себя тумбу. Тумба имеет размер $600 \times 600 \times 600$ мм., может находится либо в центре сектора, либо со смещением в пол сектора. На полигоне располагается 4 тумбы.

Полигон окружен бортом высотой 600 мм. Конфигурация полигона изменяется и определяется непосредственно перед каждым запуском робота.

На стеллаже, прилегающего одной из четырех сторон к сектору активации (старта), на высоте от 100-150 мм от уровня поверхности поля закреплен ARTag маркер (<https://goo.gl/WaTFMB>), определяющий координаты X и Y , предназначенные для обозначения финальной точки. Размер маркера - 50×50 мм. Маркер обращен лицевой стороной внутрь сектора старта данного робота. Конкретная высота расположения маркеров определяется в первый день финала и остается постоянной на все дни финального этапа. При этом допустимая погрешность установки маркеров ± 5 мм. Пример расположения маркера на стеллаже представлен на рисунке 1.2

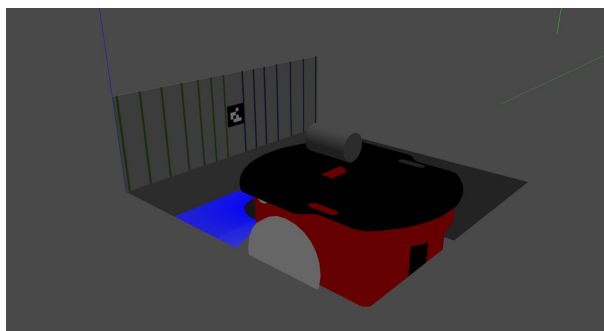


Рис. 1.2: Стеллаж с установленным ARTag маркером

ARTag метка состоит из 6×6 квадратов кодирующих биты. Черный квадрат считается равным 1, белый квадрат считается равным 0. Биты читаются слева направо сверху вниз, при этом самый первый элемент — старший бит закодированного числа. Картинка с подробным описанием приведена ниже. Элементы маркера, расположенные по его границе — всегда черные. Четыре элемента, находящиеся в углах внутреннего 4×4 квадрата определяют ориентацию маркера таким образом, что только один из них — белый. Нумерация элементов относительно ориентационных элементов обозначена на рисунке 1.3.

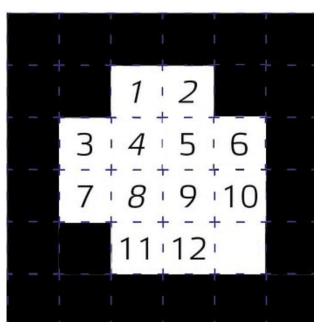


Рис. 1.3: Нумерация элементов маркера относительно ориентационных элементов

Для кодирования информации используется код с детекцией и коррекцией ошибок — код Хэмминга (<https://habr.com/ru/post/140611/>). Для кода Хэмминга используется 12 бит информации. После декодирования кода Хэмминга первые 3 бита — координата X в little endian порядке байтов, следующие 3 бита — координата Y в little endian порядке байтов, последние два бита равны 0.

- 1 Первый контрольный бит;

- 2 Второй контрольный бит;
- 3 Младший бит координаты X ;
- 4 Третий контрольный бит;
- 5 Второй бит координаты X ;
- 6 Старший бит координаты X ;
- 7 Младший бит координаты Y ;
- 8 Четвёртый контрольный бит;
- 9 Второй бит координаты Y ;
- 10 Старший бит координаты Y ;
- 11 0;
- 12 0;

Данные биты задают координаты X и Y , которые дальше будет использовать робот-погрузчик:

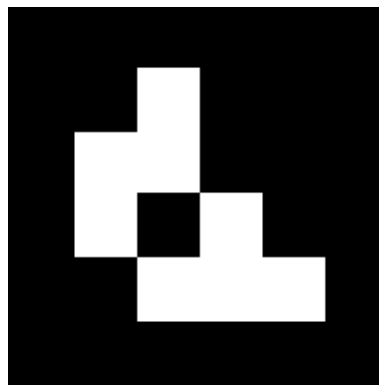


Рис. 1.4: Маркер с закодированным значением - 010011010100_2

Маркер на рисунке 1.4 кодирует число 010011010100_2 — $X = 4$, $Y = 4$.

Гарантируется, что робот начинает свое движение в секторе, к сторонам которого прилегает как минимум один стеллаж.

Сектора активаций роботов (стартов) никак не обозначаются на поле и определяются непосредственно перед каждым заездом робота.

В некоторых задачах используются координаты, в таком случае, сектор $(0,0)$ находится в левом верхнем углу, ось X направлена вправо, ось Y — вниз.

1.4. Описание конструктора

На финальном туре команды используют робота в симуляторе. Робот, собранный по дифференциальной схеме, оснащен камерой и лазерным дальномером. Дальномер позволяет получить массив данных который содержит 683 показания, полученные в диапазоне $(-120^\circ; 120^\circ)$. Показания распределены равномерно в пределах указанного диапазона. Камера установлена так, что направлена в направлении движения робота. Робот, расположенный на полигоне, управляется при помощи Python-скриптов. Для обеспечения работы робототехническое устройство использует библиотеку “**robot**”, имеющей следующие команды:

- **sleep(sec)** - пауза в sec секунд, возможно использование дробных чисел.

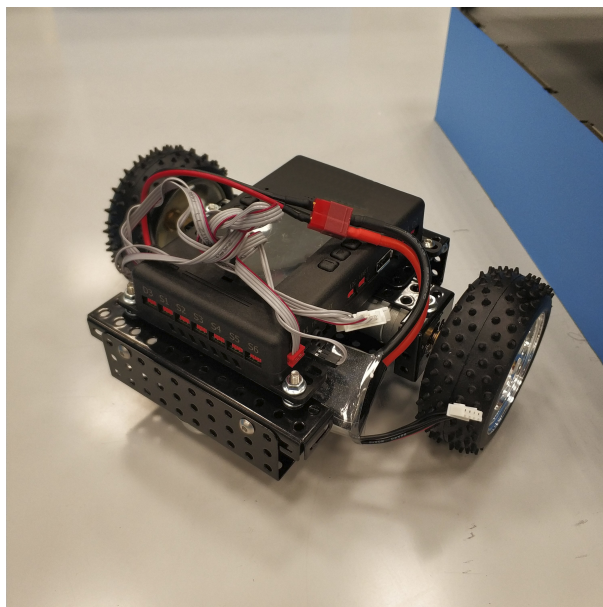


Рис. 1.5: Мобильная платформа для симулятора

- `getDirection()` - возвращает направление робота в радианах.
- `getEncoders()` - возвращает показания энкодеров в радианах в формате dict.
- `getLaser()` - возвращает показания лазера, включая время кадра, угол лазера, изменение угла между соседними показаниями и сами показания в формате dict.
- `setVelocities(linear, angular)` - подает заданные угловые и линейные скорости на робота.
- `getImage()` - возвращает изображение с камеры робота в формате OpenCV image.

1.5. Условия проведения

1. Из полученного набора датчиков команды могут выбирать те, с помощью которых, по мнению участников, можно решить задачу наиболее эффективным способом.
2. Команды могут вносить любые изменения в мобильные наземные платформы.
3. Участники во время командного этапа финального тура могут использовать интернет и заранее подготовленные библиотеки для решения задачи.
4. Участники не могут использовать помощь тренера, сопровождающего лица или привлекать третьих лиц для решения задачи.
5. Финальная задача формулируется участникам в первый день финального тура, но участники выполняют решение задачи поэтапно. Критерии прохождения каждого этапа формулируются для каждого дня финального тура. За подзадачи, решенные в конкретном этапе начисляются баллы. Баллы за подзадачи можно получить только в день, закреплённый за конкретным этапом.
6. Некоторые подзадачи строго требуют выполнения каких-то предыдущих подзадач. Выполнение данных подзадач без выполнения предыдущих допускается, однако данная попытка будет оценена в 0 баллов.

7. Во время рабочего времени команды могут проводить испытания на полигоне. Количество подходов, которое может сделать команда может быть ограничено в зависимости от ограничений, накладываемых расписанием финального этапа Олимпиады.
8. Испытания на полигоне должны осуществляться так, чтобы не мешать другим командам, проводящим в это время свои испытания на полигоне. Для этого всем командам может быть назначено ограничение по времени, которое они могут тратить на одно испытание. После истечения этого времени, команда должна дать возможность проводить испытания следующей команде.
9. Часть подзадач необходимо будет решить в симуляторе TRIK Studio: команда получает 3 тестовых виртуальных полигона с соответствующими наборами входных данных для подготовки решения, в то время как приемка решения происходит на расширенном наборе полигонов для проверки универсальности управляющей программы. Начисление баллов за подзадачи может происходить только в тот этап, в котором данные подзадачи сформулированы.
10. У команды есть неограниченное количество попыток решения подзадач в симуляторе:
 - 10.1. Решения принимаются на проверку до истечения первых 6,5 часов работы в соответствующий соревновательный день. Время может меняться в зависимости от дня.
 - 10.2. До истечения 6,5 часов, команда должна загрузить свое решение на gitlab.com в соответствующий репозиторий дня, внутри своей группы, доступ к которой участники получают в начале олимпиады.
 - 10.3. До истечения указанного времени команды могут изменять файл с решением сколько угодно раз. Проверяться будет всегда только последняя доступная версия.
 - 10.4. Зафиксировать последнее решение необходимо, создав Merge Request (MR).
 - 10.5. При создании Merge Request в качестве ответственного (assignee) укажите того, кто будет отвечать за приемку результатов.
11. Часть подзадач для реальных роботов может быть запрещена к приемке без успешного прохождения 60% всех тестов, предназначенных для проверки решения соответствующей подзадачи в симуляторе.
12. Каждый день финального тура за 2 часа (может варьироваться в зависимости от расписания) до конца выделенного рабочего времени команды должны сдать роботов в зону карантина. Время сдачи роботов в карантин может изменяться и зависит от количества команд и сложности подзадач, принимаемых в конкретный этап.
13. Перед сдачей робота в карантин команды должны загрузить на роботов управляющие программы, подготовленные для демонстрации решения задачи, а также ее копию в репозиторий, доступ к которому участники получают в начале соревновательного дня. Без программы, загруженной в репозиторий, команды не допускаются до проверки решения на реальном роботе.
14. Зафиксировать загрузку файлов, содержащих решения задачи на реальном роботе, необходимо при помощи MR.
15. Программа должна успешно собираться в pdf.
16. После момента, когда все роботы сданы в карантин, судьи по одной вызывают команды для приемки решения подзадач, закрепленных за этапом конкретного дня финального тура.

17. Может быть предусмотрено до двух попыток сдачи решения одной и той же подзадачи на реальных роботах. Конкретное количество попыток определяется в конкретных подзадачах.
18. После прохождения приемочных запусков, баллы набранные командой заносятся судьями в протокол. Один из участников команды расписывается за набранный результат, подтверждая согласие команды с оценкой проведенных запусков.
19. Роботы должны выполнять задание полностью автономно. Удаленное управление не допускается. Касание какого-либо робота участником команды после его старта во время приемочных запусков не допускается. Алгоритм, реализующий систему управления группой роботов, должен планировать свое выполнение, полагаясь только на информацию с датчиков.
20. Введение данных в программу до старта устройства (например, координат робота в начале работы) разрешается только для тех задач, где это явно прописано. Во всех других случаях введение данных в программу роботов перед запуском запрещено.
21. Для всех роботов программа должна быть одинаковой, допускается отличие лишь в разрешенных входных данных.
22. Если какая-то подзадача подразумевает считывание информации с элементов, расположенных на полигоне, запрещается при запуске роботов вводить информацию о положении этих элементов или значениях, которые данные элементы определяют.
23. Если во время приемочных запусков у судьи возникли сомнения о том, что задачи подэтапа решены корректно (роботы не выполняют задачу полностью автономно, участник вводит значения в каких-либо роботов перед запуском), то он вправе провести инспекцию кода. По результатам инспекции, судья вправе снять с команды баллы, набранные за данный этап.
24. Если во время приемочных запусков у судьи возникает ситуация, когда он не может однозначно решить выполняются ли критерии решения подзадачи, он вправе принять решение не в пользу команды.
25. В случае если возникает техническая проблема, независящая от участников, то по решению судей может быть предоставлена возможность перезапуска
26. Команда вправе обсуждать с судьей результаты приемочных запусков до вызова следующей команды, но финальное решение остается о начислении баллов остается за судьей.

1.6. Процедура проведения приемочных запусков и критерии оценки

Первый этап

1. Командам необходимо подготовить две задачи для симулятора:
 - 1.1. В качестве первой задачи участникам необходимо выполнить следующее:
 - 1.1.1. Робот устанавливается в модели логистического центра в заранее неизвестном секторе. Задача робота выполнить все команды переданные через входной файл. Робот выполнил все команды и вывел

finish на всех проверочных полигонах.

Входные данные: через файл `input.txt` управляющей программе передаются:

А. В первой строке через пробел — команды которые необходимо выполнить, где:

- 0 Действие не требуется (“N”);
- 1 Роботу необходимо повернуть налево на 90° внутри данного сектора (“L”);
- 2 Роботу необходимо повернуть направо на 90° внутри данного сектора (“R”);
- 3 Роботу необходимо проехать в следующий по направлению движения сектор (“F”);

Ожидаемый результат: После запуска программы робот выполнил все команды, остановился и вывел на экран **finish**.

1.2. В качестве второй задачи:

1.2.1. Робот устанавливается в модели логистического центра в заранее неизвестном секторе. Задача робота проехать из точки старта в точку финиша. Робот проехал из точки старта в точку финиша и вывел **finish** на всех проверочных полигонах. Координаты сектора старта, включая направление и финиша передаются через входной файл. Структура лабиринта не известна.

Входные данные: через файл `input.txt` управляющей программе передаются:

А. В первой строке через пробел — координаты старта X_s, Y_s , направление старта D_s ($0 \leq X_s, Y_s \leq 7, D_s \in [0, 3]$):

- 0 Робот направлен вверх (в сторону отрицательного направления оси Y);
- 1 Робот направлен направо (в сторону положительного направления оси X);
- 2 Робот направлен вниз (в сторону положительного направления оси Y);
- 3 Робот направлен налево (в сторону отрицательного направления оси X);

В. Во второй строке через пробел — координаты финиша X_f, Y_f , $0 \leq X_f, Y_f \leq 7$;

Ожидаемый результат: После запуска программы робот доехал до финиша, остановился и вывел на экран **finish**.

1.3. Правила именования файлов с управляющей программой для проверки решений в симуляторе:

1.3.1. Для первой задачи: `sim_1.js`;

1.3.2. Для второй задачи: `sim_2.js`.

2. Конфигурация робота в симуляторе следующая:

2.1. Конфигурация моторов:

2.1.1. М3 — правый мотор;

2.1.2. М4 — левый мотор.

2.2. Конфигурация датчиков:

2.2.1. D1 — ультразвуковой датчик, направленный вперёд;

- 2.2.2. D2 — ультразвуковой датчик, направленный назад;
 - 2.2.3. A1 — инфракрасный датчик, направленный вправо;
 - 2.2.4. A2 — инфракрасный датчик, направленный влево.
3. Команде необходимо будет подготовить решения для двух разных подзадач для реальных роботов. На демонстрацию каждого решения предоставляется 2 попытки.
 4. Все попытки осуществляются 18 марта.
 5. За 5 мин до сдачи в карантин для 1ой подзадачи судья определяет количество и набор команд для выполнения роботом.
 6. После сдачи в карантин для 1ой подзадачи судья определяет сектор старта и направление робота в секторе старта.
 7. За 5 мин до сдачи в карантин для 2ой подзадачи судья определяет сектор старта, направление старта и сектор финиша для робота. Данные значения команда должна внести в программу перед тем, как сдать робота в карантин.
 8. Секторы старта могут быть различными в разных попытках. Данные значения команда должна внести в программу перед тем, как сдать робота в карантин.
 9. Правила именования файлов с программой управления:
 - 9.1. для первой попытки первой подзадачи: `real_1_1.js`;
 - 9.2. для второй попытки первой подзадачи: `real_1_2.js`;
 - 9.3. для первой попытки второй подзадачи: `real_2_1.js`;
 - 9.4. для второй попытки второй подзадачи: `real_2_2.js`.
 10. Merge Request необходимо назвать следующим образом: “код команды_day1”. Например: “irs202000_day1”.
 11. Максимальное время выполнения одной попытки для 1ой подзадачи — 2 минуты.
 12. Максимальное время выполнения одной попытки для 2ой подзадачи — 5 минут.
 13. Баллы за решение задач этапа:
 - 13.1. **Первая задача в симуляторе:** робот смог выполнить всю задачу на всех проверочных полигонах — 12 баллов.
 - 13.2. **Вторая задача в симуляторе:** робот смог выполнить всю задачу на всех проверочных полигонах — 20 баллов.
 - 13.3. **Первая подзадача на реальном роботе:** Робот устанавливается в модели логистического центра в заранее неизвестном секторе. Задача робота выполнить все команды. Робот выполнил все команды и вывел `finish`. — 16 баллов.
 - 13.4. **Вторая подзадача на реальном роботе:** Робот устанавливается в модели логистического центра в заранее неизвестном секторе. Задача робота проехать из точки старта в точку финиша. Робот проехал из точки старта в точку финиша и вывел `finish`. Координаты сектора старта, включая направление и финиша передаются через входной файл. Структура лабиринта не известна. — 20 баллов.
 14. Выводить `finish` следует не менее 10 секунд.
 15. Баллы за первую подзадачу не начисляются, если не было частично засчитано решение первой задачи в симуляторе
 16. Баллы за вторую подзадачу не начисляются, если не было частично засчитано решение второй задачи в симуляторе.

17. Баллы за все попытки в каждой подзадаче суммируются.
18. Выполнение всех критериев в каждой из двух попыток всех двух подзадач дает дополнительные 8 баллов.
19. Максимальное количество баллов за этап — 112.

Второй этап

1. Командам необходимо подготовить две задачи для симулятора:

- 1.1. В качестве первой задачи участникам необходимо выполнить следующее:

- 1.1.1. Робот устанавливается в модели логистического центра в заранее неизвестном секторе. Необходимо в процессе движения определить своё местоположение, остановиться и вывести на экран координаты робота в формате "(X, Y)" без пробелов и без кавычек. Направление старта передаётся через входной файл.

Входные данные: через файл `input.txt` управляющей программе передаются:

- A. В первой строке — направление старта D_s :

- 0 Робот направлен вверх (в сторону отрицательного направления оси Y);
- 1 Робот направлен направо (в сторону положительного направления оси X);
- 2 Робот направлен вниз (в сторону положительного направления оси Y);
- 3 Робот направлен налево (в сторону отрицательного направления оси X);

Ожидаемый результат: После запуска программы робот определил своё местоположение, остановился и вывел на экран `finish`.

- 1.2. В качестве второй задачи:

- 1.2.1. Робот устанавливается в модели логистического центра в заранее неизвестном секторе. Известно что один сектор данного логистического центра заблокирован другим роботом. Координаты этого сектора передаются через входной файл. Необходимо в процессе движения определить своё местоположение и приехать в сектор, смежный с сектором нахождения другого робота и вывести `finish`. Структура лабиринта заранее не известна.

Входные данные: через файл `input.txt` управляющей программе передаются:

- A. В первой строке через пробел — направление старта D_s

- 0 Робот направлен вверх (в сторону отрицательного направления оси Y);
- 1 Робот направлен направо (в сторону положительного направления оси X);
- 2 Робот направлен вниз (в сторону положительного направления оси Y);
- 3 Робот направлен налево (в сторону отрицательного направления оси X);

- B. Во второй строке через пробел — координаты расположения

другого робота X, Y ($0 \leq X_s, Y_s \leq 7$);

Ожидаемый результат: После запуска программы робот доехал до одного из смежных секторов с сектором нахождения другого робота, остановился и вывел на экран **finish**.

- 1.3. Правила именования файлов с управляющей программой для проверки решений в симуляторе:
 - 1.3.1. Для первой задачи: **sim1.js**;
 - 1.3.2. Для второй задачи: **sim2.js**.
2. Конфигурация робота в симуляторе следующая:
 - 2.1. Конфигурация моторов:
 - 2.1.1. М3 — правый мотор;
 - 2.1.2. М4 — левый мотор.
 - 2.2. Конфигурация датчиков:
 - 2.2.1. D1 — ультразвуковой датчик, направленный вперёд;
 - 2.2.2. D2 — ультразвуковой датчик, направленный назад;
 - 2.2.3. A1 — инфракрасный датчик, направленный вправо;
 - 2.2.4. A2 — инфракрасный датчик, направленный влево.
3. Команде необходимо будет подготовить решения для одной задачи для реальных роботов. На демонстрацию решения предоставляется 2 попытки.
4. Все попытки осуществляются 19 марта.
5. Merge Request необходимо назвать следующим образом: “код команды_day2”. Например: “irs202000_day2”.
6. За 10 мин до сдачи в карантин судья определяет направление старта для робота. Данное значение команда должна внести в программу перед тем, как сдать робота в карантин.
7. Направление старта могут быть различными в разных попытках. Данные значения команда должна внести в программу перед тем, как сдать робота в карантин.
8. После сдачи в карантин судья определяет секторы старта обоих роботов.
9. Правила именования файлов с программой управления:
 - 9.1. для первой попытки задачи для реального робота: **real_1_1.js**;
 - 9.2. для второй попытки задачи для реального робота: **real_1_2.js**;
10. Максимальное время выполнения одной попытки - 7 минут.
11. Баллы за решение задач этапа:
 - 11.1. **Первая задача в симуляторе:** робот смог определить своё местоположение на всех проверочных полигонах — 16 баллов.
 - 11.2. **Вторая задача в симуляторе:** робот смог достичь смежного сектора на всех проверочных полигонах — 20 баллов.
 - 11.3. **Задача на реальном роботе:** Роботы располагаются в случайных секторах робототехнического полигона.
 - 11.3.1. Первый робот определил своё местоположение на поле, остановился, издал звуковой сигнал, вывел на экран свои координаты в формате (X, Y) . — 14 баллов.
 - 11.3.2. Координаты второго робота также определены. Второй робот издал звуковой сигнал, вывел на экран свои координаты в формате (X, Y) . — 10 баллов.

- 11.3.3. Во время решения поставленной задачи второй робот не покидал изначальных сектор. — 8 баллов.
12. Выводить значение метки следует не менее 10 секунд.
13. Баллы за задачу на реальном роботе не начисляются, если не было частично засчитано решение первой задачи в симуляторе.
14. Баллы за все попытки на реальном роботе суммируются.
15. Выполнение всех критериев в каждой из двух попыток всех двух подзадач дает дополнительные 8 баллов.
16. Максимальное количество баллов за этап — 108.

Третий этап

1. В качестве задачи для симулятора участникам необходимо выполнить следующее:
- 1.1. Робот устанавливается в модели логистического центра в заранее неизвестном секторе. При этом структура логистического центра не известна заранее. Задача робота проехать из точки старта в точку финиша, согласно маршруту, заданному изображением ARTag маркера, заранее считанным с камеры реального устройства. На финише необходимо вывести на экран слово **finish**.
Входные данные: через файл `input.txt` управляющей программе передаются:
- В первой строке 19200, разделенных пробелом, целых чисел $P_{1,i}$ ($0 \leq P_{1,i} \leq 2^{24}$) — изображение ARTag маркера;
- Каждое число в маркере — точка, закодированная в формате RGB, т.е. строка с изображением маркера эквивалентна снимку разрешением 160×120 точек.
- Ожидаемый результат:* После запуска программы робот доехал до финиша, остановился и вывел на экран **finish**.
- 1.2. Имя файла с управляющей программой для проверки решения в симуляторе: `sim1.js`.
2. Конфигурация робота в симуляторе следующая:
- 2.1. Конфигурация моторов:
- 2.1.1. M3 — правый мотор;
 - 2.1.2. M4 — левый мотор.
- 2.2. Конфигурация датчиков:
- 2.2.1. D1 — ультразвуковой датчик, направленный вперед;
 - 2.2.2. D2 — ультразвуковой датчик, направленный назад;
 - 2.2.3. A1 — инфракрасный датчик, направленный вправо;
 - 2.2.4. A2 — инфракрасный датчик, направленный влево.
3. Команде необходимо будет подготовить решения для двух разных подзадач для реального робота. На демонстрацию каждого решения предоставляется 2 попытки.
4. Все попытки осуществляются 20 марта.
5. Merge Request необходимо назвать следующим образом: “код команды_day3”. Например: “irs202000_day3”.

6. Допускается возможность демонстрации первой подзадачи во время периода отладки.
7. Сектор старта может быть разным для каждой попытки.
8. Правила именования файлов с программой управления:
 - 8.1. для первой подзадачи: `real_1.js`;
 - 8.2. для второй подзадачи: `real_2.js`;
9. Максимальное время выполнения одной попытки - 3 минуты.
10. Баллы за решение задач этапа:
 - 10.1. **Задача в симуляторе:** робот проехал из точки старта в точку финиша на всех проверочных полигонах — 16 баллов.
 - 10.2. **Первая подзадача на реальном роботе:** Робот распознал верно ARTag метку, которая установлена прямо перед камерой, и вывел на экран верные команды, закодированные в ARTag метке. Допускается ручная настройка (наведение) камеры на маркер (включая визуальную корректировку средствами просмотра). — 8 баллов.
 - 10.3. **Вторая подзадача на реальном роботе:** Робот располагается в секторе старта, с неизвестной стороны которого располагается ARTag метка. Робот нашел метку, издал звуковой сигнал и вывел на экран верные команды, закодированные в ARTag метке — 12 баллов.
11. Цифры, означающие команды, необходимо выводить на экран через пробел. Допускается вывод в несколько строк.
12. За вторую подзадачу начисляется только половина возможных баллов, если не было засчитано решение в симуляторе, а также если не сдана первая подзадача.
13. Выводить значение метки следует не менее 10 секунд.
14. Баллы за все попытки в каждой подзадаче суммируются.
15. Выполнение всех критериев в каждой из двух попыток всех двух подзадач дает дополнительные 4 балла.
16. Максимальное количество баллов за этап — 60.

Четвёртый этап

1. В качестве задачи для симулятора участникам необходимо выполнить следующее:
 - 1.1. Робот устанавливается в модели логистического центра в заранее неизвестном секторе. Известно что один сектор данного логистического центра заблокирован другим роботом. Необходимо в процессе движения определить своё местоположение, доехать в сектор старта и вывести на экран координаты одного из достижимых смежных секторов заблокированного робота в формате " (X, Y) " без пробелов и без кавычек.
Входные данные: через файл `input.txt` управляющей программе передаются:
 - 1.1.1. В первой строке — направление старта D_s :
 - 0 Робот направлен вверх (в сторону отрицательного направления оси Y);
 - 1 Робот направлен направо (в сторону положительного направления оси X);
 - 2 Робот направлен вниз (в сторону положительного направле-

- ния оси Y);
- 3 Робот направлен налево (в сторону отрицательного направления оси X);
- 1.1.2. Во второй строке через пробел — координаты расположения второго робота X, Y ($0 \leq X, Y \leq 7$);
- Ожидаемый результат:* После запуска программы робот определил своё местоположение, вернулся в сектор старта и вывел на экран координаты.
- 1.2. Имя файла с управляющей программой для проверки решения в симуляторе: `sim1.js`.
2. Конфигурация робота в симуляторе следующая:
- 2.1. Конфигурация моторов:
- 2.1.1. M3 — правый мотор;
- 2.1.2. M4 — левый мотор.
- 2.2. Конфигурация датчиков:
- 2.2.1. D1 — ультразвуковой датчик, направленный вперёд;
- 2.2.2. D2 — ультразвуковой датчик, направленный назад;
- 2.2.3. A1 — инфракрасный датчик, направленный вправо;
- 2.2.4. A2 — инфракрасный датчик, направленный влево.
3. Команде необходимо будет подготовить решения для одной подзадачи для реальных роботов. На демонстрацию решения предоставляется 1 попытка.
4. Все попытки осуществляются 21 марта.
5. Merge Request необходимо назвать следующим образом: “код команды_day4”. Например: “irs202000_day4”.
6. Стартовым направлением для первого робота является направление 0.
7. После сдачи в карантин для подзадачи судья определяет секторы старта и направление старта 2го робота.
8. Правила именования файлов с программой управления:
- 8.1. для первой попытки подзадачи: `real_1_1.js`;
9. Максимальное время выполнения одной попытки - 15 минут.
10. Баллы за решение задач этапа:
- 10.1. **Задача в симуляторе:** робот смог выполнить всю задачу на всех проверочных полигонах — 20 баллов.
- 10.2. **Задача на реальном роботе:** Роботы располагаются в секторах старта, задача определить свое местоположение, выполнить вторым роботом команды (относительно начального направления робота), переданные через ARTag метку, расположенную на случайном стеллаже вокруг сектора старта, доехать первым роботом до второго и после до любого сектора старта.
- 10.2.1. Второй робот успешно считал ARTag метку, выполнил команды, остановился, издал звук и вывел на экран *finish* и более не двигался. — 24 балла.
- 10.2.2. Первый робот локализовался, доехал до смежного сектора, издал звук. — 24 балла.
- 10.2.3. Первый робот доехал до смежного сектора, после перемещения второго робота — 8 баллов.
- 10.2.4. Первый робот локализовался, доехал до смежного сектора, издал

звук. Через 10 сек продолжил движение до любого сектора старта, остановился, издал звук и вывел на экран *finish*.—24 балла.

10.2.5. Первый робот доехал до ближайшей (с точки зрения количества секторов) зоны старта. Перед этим, первый робот доехал до смежного сектора после перемещения второго робота — 16 баллов.

11. Баллы за задачу на реальном роботе не начисляются, если не было частично засчитано решение задачи в симуляторе.
12. Частичным решением в симуляторе считается успешное прохождение 30% всех тестов.
13. Во время решения задачи возможен перезапуск. При перезапуске команда может поправить своё решение, потратив на это **не более трёх минут**.
14. При перезапуске набранные баллы не пропадают, время не останавливается.
15. В случае если команда использовала перезапуск, то она получит половину из достигнутых баллов за всё попытку.
16. Баллы за все попытки в каждой подзадаче суммируются.
17. Выполнение всех критериев(не обязательно на максимум баллов) в каждой из двух попыток задачи на реальном роботе даёт дополнительные 4 балла.
18. Максимальное количество баллов за этап — 120.

1.7. Критерии определения команды-победителя командного тура

1. Максимальный балл за командный этап — 400 баллов.
2. Сумма баллов, набранных за решения подзадач всех этапов командной части финального этапа, определяет итоговую результативность команды (измеряемую в баллах).
3. Команды ранжируются по результативности.
4. Команда победитель определяется, как команда с максимальной результативностью.