

1. ФИНАЛЬНЫЙ ЭТАП

Версия от 24 марта 2021 г.

Задача командного тура

1.1. Легенда

В будущем в автоматических логистических центрах почти всю работу будут выполнять роботы, а не люди. Таким образом, будет минимизироваться человеческий фактор, а вслед за этим повышаться скорость и точность работы. Этими роботами будет управлять интеллектуальное программное обеспечение по постановке и распределению задач.

В таких логистических центрах в любой момент может произойти нештатная ситуация и система из роботов под управлением программного обеспечения должна уметь справляться с такими ситуациями. Для того, чтобы такая «красивая картинка» сложилась в реальности, нужны команды опытных разработчиков.

Поэтому задача заключительного этапа профиля «Интеллектуальные робототехнические системы» основывается на робототехнической задаче SLAM — одновременной локализации и картографировании.

Она будет требовать от участников применения таких навыков, как определение объектов окружающей обстановки робота с помощью камеры, построение карты с помощью дальномеров, выбор стратегии по сбору недостающих данных об окружающем пространстве через планирование перемещений робота.

Несмотря на то, что, по легенде, описанные выше задачи ставятся перед складским роботом, подобные алгоритмы используются также беспилотными автомобилями для успешного перемещения в потоке машин и прибытия в пункт назначения, роботами-курьерами, роботизированной сельскохозяйственной техникой, роботами-пылесосами для качественной чистки обслуживаемых помещений.

Задача заключительного этапа будет следующей: В логистическом центре произошла перезагрузка всех систем, что привело к сбросу информации о местоположении работающего в данный момент робота-погрузчика, а также карты данной части логистического центра.

В начале выполнения задания считается, что робототехническое устройство активируется в правом нижнем углу логистического центра. Робот должен локализоваться, найти ближайшую к стене тумбу и определить своё глобальное местоположение. Далее он должен найти ARTag метку, по ней узнать местоположение сервисного обслуживания и приехать на эти координаты. Структура логистического центра

неизвестна заранее. При перемещении робот не должен повреждать логистический центр.

Задача участников Олимпиады — разработать программу управления робототехническим устройством для выполнения задания, описанного выше.

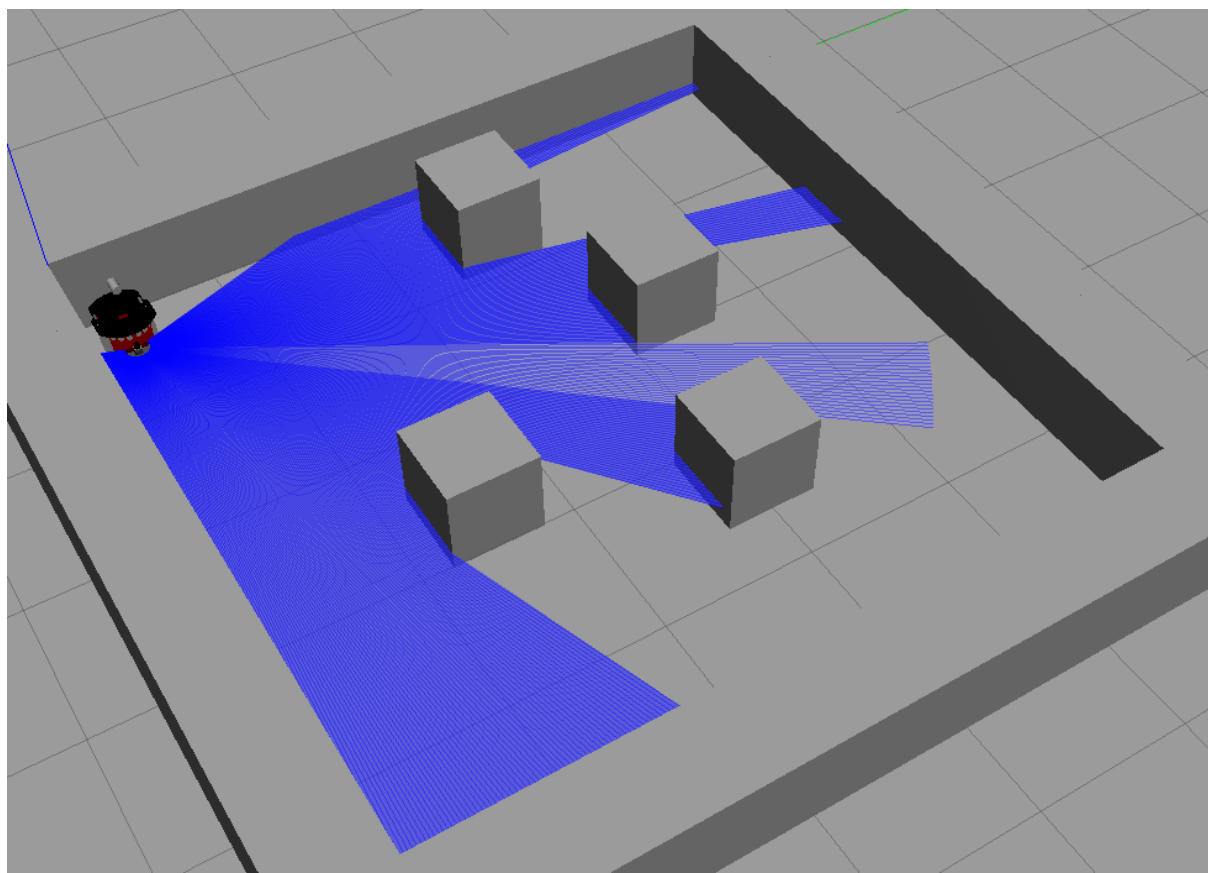


Рис. 1.1: Полигон для запуска робототехнических устройств на финале Олимпиады НТИ

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 правом нижнем углу полигона с неизвестной ориентацией. На полигоне имеются четыре тумбы различных цветов. Цвет

тумбы, наименее удаленной от стены полигона, задаёт направления осей координат и, соответственно, начальные координаты. Тумбы располагаются параллельно стенкам и координаты их центра задаются следующим образом: $(0, 5i; 0, 5j)$, где i, j — целые числа (т.е. тумбы могут быть смещены на пол сектора). В процессе работы робота необходимо определить свои глобальные координаты. *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 с неизвестной ориентацией начинает своё движение рядом с желтой тумбой с случайной её стороны. Необходимо найти ARTag маркер и расшифровать глобальные 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.
- Декодирование бинарного кода с использованием кода Хэмминга.

Третий этап

Задача: Робот начинает свое движение в правом нижнем углу полигона с неизвестной ориентацией. 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 конечного сектора. Тумбы располагаются параллельно стенкам и координаты их центра задаются следующим образом: $(0, 5i; 0, 5j)$, где i, j — целые числа (т.е. тумбы могут быть смещены на пол сектора). Робота необходимо доехать до конечного 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 $600 \times 600 \times 400$ мм (Д \times Ш \times В), может находиться либо в центре сектора, либо со смещением в половину сектора. Координаты центра тумбы можно представить следующим образом: $(0, 5i; 0, 5j)$, где i, j — целые числа. На полигоне располагается 4 тумбы.

Полигон окружен бортом высотой 400 мм. Конфигурация полигона изменяется и определяется непосредственно перед каждым запуском робота.

Тумба жёлтого цвета, расположенная на полигоне содержит ARTag маркер, расположенный на одной из боковых сторон, на высоте 250-350 мм от основания. Горизонтально маркер расположен по центру тумбы.

Он определяет глобальные координаты X и Y — координаты местоположения сектора сервисного обслуживания, служащего финальной точкой. Размер маркера - 50×50 мм. 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 относительно их первичной установки. Пример расположения маркера на стеллаже представлен на рисунке 1.2

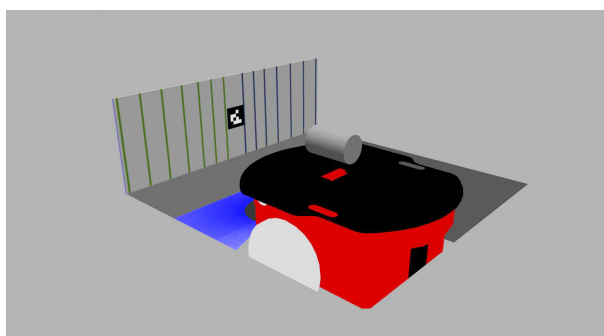


Рис. 1.2: Тумба с установленным ARTag маркером

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.

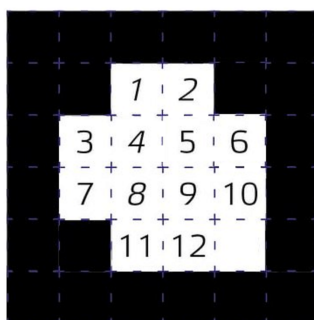


Рис. 1.3: Нумерация элементов маркера относительно ориентационных элементов

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.

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 битов, следующие 3 бита — координата Y в little endian порядке битов, последние два бита равны 0.

- 1 Первый контрольный бит;
- 2 Второй контрольный бит;
- 3 Младший бит координаты X ;
- 4 Третий контрольный бит;
- 5 Средний бит координаты X ;
- 6 Старший бит координаты X ;
- 7 Младший бит координаты Y ;
- 8 Четвёртый контрольный бит;
- 9 Средний 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:

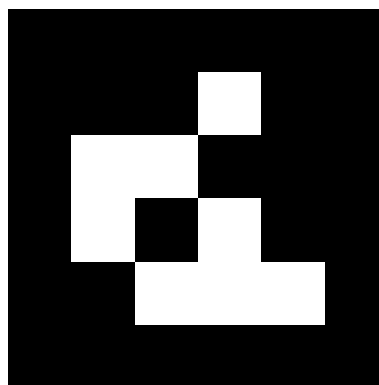


Рис. 1.4: Маркер с закодированным значением - 100011010100_2

Маркер на рисунке 1.4 кодирует число 100011010100_2 — $X = 110_2 = 6_{10}$, $Y = 100_2 = 4_{10}$.

Робот начинает свое движение в правом нижнем секторе. Гарантируется, что координаты сервисного обслуживания достижимы. Сектора активации робота (старта) никак не обозначаются на поле и определяются непосредственно перед каждым заездом робота.

В некоторых задачах используются локальные (относительные) координаты. Эти координаты зависят от направления старта робота и определяются следующим образом: сектор $(0, 0)$ находится в точке начального расположения робота, ось Y направлена вдоль начального направления робота, ось X — направлена вправо отно-

сительно направления старта робота.

Также существует 4 вида глобальных координат: жёлтые, красные, синие, зелёные. Во время попытки активным может быть только один вид, это определяется цветом тумбы, ближайшей к стенке. Гарантируется, что глобальные координаты можно определить однозначно в процессе движения. Ориентация и начало отсчета координат представлены на рис. 1.5

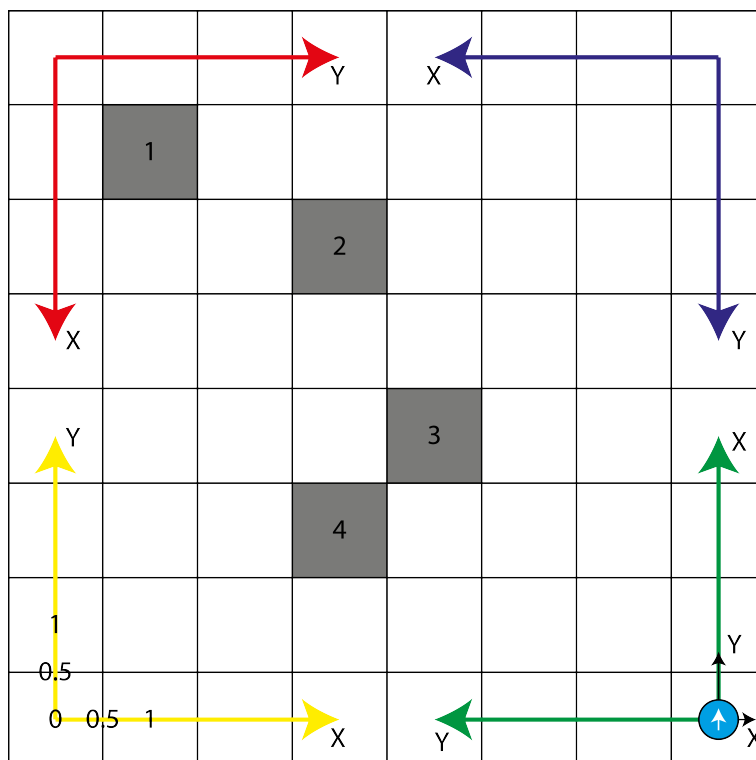


Рис. 1.5: Расположение и ориентация глобальных координат

Также на рис. 1.5 представлен пример расположения тумб и робота. Робот — голубой круг, белая стрелка — направление старта робота. Чёрная ось координат означает относительные координаты. Например, тумба 1 имеет следующие локальные координаты: $(-6; 6)$. Также эта тумба является ближайшей к стенке, следовательно, её цвет кодирует вид глобальных координат. Рассмотрим следующие варианты, если цвет тумбы 1:

1. Жёлтый, то глобальные координаты данной тумбы: $(1; 6)$;
2. Красный, то глобальные координаты данной тумбы: $(1; 1)$;
3. Синий, то глобальные координаты данной тумбы: $(6; 1)$;
4. Зелёный, то глобальные координаты данной тумбы: $(6; 6)$;

1.4. Описание робота

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 направлена «вперед» по направлению движения робота. Разреше-

ние камеры: 1280×720 . Робот, расположенный на полигоне, управляется при помощи Python-скриптов. Для обеспечения работы робототехническое устройство использует библиотеку “robot”, имеющую следующие команды:

- `sleep(sec)` - пауза в sec секунд, возможно использование дробных чисел.
- `time()` -возвращает текущее время в симуляции (simulation time в Gazebo) в секундах.
- `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)` - подать заданные угловые и линейные скорости на робота.
- `getImage()` - возвращает RGB-изображение с камеры робота в формате OpenCV image.

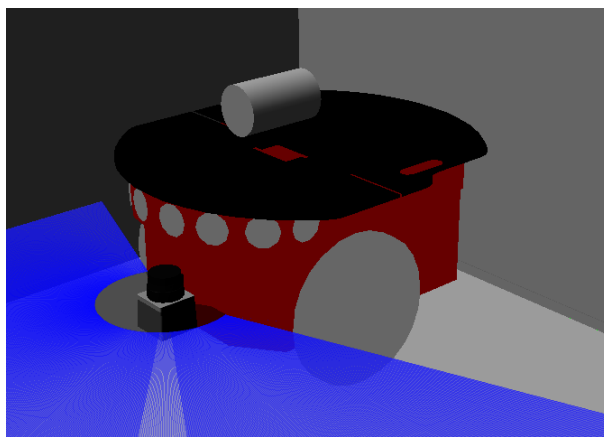


Рис. 1.6: Мобильная платформа для симулятора

1.5. Условия проведения

1. Из имеющегося набора датчиков команды могут выбирать те, с помощью которых, по мнению участников, можно решить задачу наиболее эффективным способом.
2. Участники во время командного этапа финального тура могут использовать интернет и заранее подготовленные библиотеки для решения задачи.
3. Участники могут использовать предустановленные библиотеки `numpy` и `opencv`. Использование иных библиотек, помимо стандартных, не допускается. Их использование может повлечь аннулирование баллов за задачу.
4. Использование `gos` функций и библиотек, включая `gospu` не допускается. Исключением является библиотека “robot” и иные команды созданные организаторами.
5. Для проведения финала подготовлен образ системы (доступный по ссылке: <http://bit.ly/gazebo-image-2>), с настроенным программным обеспечением. Участники должны использовать данный образ.

6. Допускается использование и настройка иных систем. В этом случае вся ответственность за настройку и работоспособность лежит на участниках. Также необходимо убедиться в работоспособности решения в виртуальной системе, выданной организаторами.
7. Проверка решений будет происходить в виртуальной системе или в системе с подобными зависимостями, например, с такими же версиями используемых библиотек.
8. Участники не могут использовать помощь тренера, сопровождающего лица или привлекать третьих лиц для решения задачи.
9. 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.
10. Во время рабочего времени команды могут проводить испытания в симуляторе Gazebo: команда получает не менее 3 тестовых виртуальных полигонов с соответствующими наборами входных данных для подготовки решения, в то время как прием решений происходит на иных полигонах, составленных для проверки универсальности управляющей программы. Для запуска данных тестов рекомендуется использовать следующую команду в терминале: `./sh/run_simulation.sh N`, где N – номер теста.
11. Также в директории `open_worlds` вашего репозитория могут находиться комментарии к открытым тестам, содержащие ответ на задачу или подсказки к ним.
12. Каждый день финального тура в 15:00 по МСК (может варьироваться в зависимости от расписания) команда должна загрузить свое решение на `gitlab.com` в соответствующий репозиторий дня, внутри своей группы, доступ к которой участники получают в начале олимпиады. Время может изменяться и зависит от количества команд и сложности подзадач, принимаемых в конкретный этап.
13. До истечения указанного времени команды могут изменять файл с решением сколько угодно раз. Проверяться будет всегда только последняя доступная версия.
14. Необходимо зафиксировать последнее решение - создать Merge Request (MR).
15. При создании Merge Request в качестве ответственного (assignee) укажите того, кто будет отвечать за приемку результатов.
16. После момента, когда все отправили свои решения на проверку, судьи приступают к проверке отправленных решений и подзадач, закрепленных за этапом конкретного дня финального тура.
17. Может быть предусмотрено до двух попыток сдачи решения одной и той же подзадачи в симуляторе. Конкретное количество попыток определяется в конкретных подзадачах.
18. После прохождения приемочных запусков, баллы, набранные командой, заносятся судьями в протокол.
19. Робот должен выполнять задание полностью автономно. Удаленное управление не допускается.
20. Не допускается вывод в консоль ничего, кроме информации, требуемой в задаче. В случае вывода иной информации, она может быть расценена как часть

ответа.

21. 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.
22. В случае, если возникает техническая проблема, независящая от участников, то по решению судей может быть предоставлена возможность перезапуска.
23. Результат тестирования участникам будет сообщен на следующий день после сдачи решения.
24. Команды могут до 10:00 МСК обратиться с апелляцией. Жюри вправе провести тестирование или видео воспроизведение попытки (если таковая имеется) и назначить баллы за соответствующие подзадачи.

1.6. Процедура проведения приемочных запусков и критерии оценки

Первый этап

1. Командам необходимо подготовить следующую задачу для симулятора:
 - 1.1. Робот начинает свое движение в правом нижнем углу полигона с неизвестной ориентацией. На полигоне имеются четыре тумбы различных цветов. Цвет тумбы наименее удаленной от стены полигона задаёт направления осей координат и, соответственно, начальные координаты. Тумбы располагаются параллельно стенкам и координаты их центра задаются следующим образом: $(0, 5i; 0, 5j)$, где i, j — целые числа (т.е. тумбы могут быть смещены на пол сектора). В процессе работы роботу необходимо определить свои глобальные координаты.
2. Основной файл с управляющей программой для проверки решения должен называться `main.py` и находиться в корне репозитория.
3. Merge Request необходимо назвать следующим образом: “код команды_day1”. Например: “irs202100_day1”.
4. Максимальное время выполнения одной попытки для задачи — 5 минут.
5. Решение будет проверяться на 2 проверочных полигонах.
6. Баллы за решение подзадач этапа:
 - 6.1. Проекция всего робота покинула сектор старта — 4 балла.
 - 6.2. Робот смог определить расстояние между стенкой и ближайшей тумбой и вывел его в консоль в формате: `distance X` — 11 баллов.
 - 6.3. Робот определил цвет ближайшей к стенке тумбы и вывел его в консоль в формате: `color X`, где X — цвет (red, green, blue, yellow) — 13 баллов.
 - 6.4. После определения цвета или расстояния робот остановился и вывел в консоль относительные координаты в формате: `temp coordinates X Y` и не менее чем через 10 сек продолжил движение — 15 баллов.
 - 6.5. Робот смог определить свои точные координаты и вывел их в консоль в формате: `coordinates X Y`, где X и Y — координаты в секторах. Робот продолжил движение не менее чем через 10 сек (если необходимо). — 22 балла.

7. Значение координат необходимо вычислять для центра робота.
8. Значение координат необходимо вывести с точностью до целых.
9. Баллы за все попытки в каждой подзадаче суммируются.
10. Выполнение всех критериев в каждой из двух попыток всех подзадач дает дополнительные 10 баллов.
11. Максимальное количество баллов за этап — 140.

Второй этап

1. Командам необходимо обновить библиотеку для корректной работы `robot.sleep` функции в симуляторе. Это позволит командам работать с временем из симулятора, а не реальным. Проверка решений будет происходить на обновлённом рабочем окружении. Для обновления следует воспользоваться следующей командой: `./sh/update.sh`
2. Командам необходимо подготовить следующую задачу для симулятора:
 - 2.1. Робот с неизвестной ориентацией начинает своё движение рядом с жёлтой тумбой с случайной её стороны. Необходимо найти ArTag маркер и расшифровать глобальные координаты, закодированные на нем.
3. Основной файл с управляющей программой для проверки решения должен называться `main.py` и находиться в корне репозитория.
4. Merge Request необходимо назвать следующим образом: “код команды _day2”. Например: “irs202100_day2”.
5. Максимальное время выполнения одной попытки для задачи — 3 минуты.
6. Решение будет проверяться на 2 проверочных полигонах.
7. Баллы за решение подзадач этапа:
 - 7.1. Робот смог найти жёлтую тумбу, остановился, повернулся в направлении данной тумбы, вывел в консоль `found` и через 10 сек при необходимости продолжил движение — 4 балла.
 - 7.2. Робот смог найти сторону с маркером, остановился и вывел в консоль: `found marker` и через 10 сек при необходимости продолжил движение — 11 баллов.
 - 7.3. Робот смог распознать сырое значение маркера (все биты в двоичном виде) и вывел в консоль данные значения в формате: `binary X` — 16 баллов.
 - 7.4. Робот смог распознать значения координат и вывел в консоль координаты в формате: `coordinates X Y` — 20 баллов.
8. Значение координат необходимо вывести с точностью до целых.
9. Баллы за все попытки в каждой подзадаче суммируются.
10. Выполнение всех критериев в каждой из двух попыток всех подзадач дает дополнительные 8 баллов.
11. Максимальное количество баллов за этап — 110.

Третий этап

1. Командам необходимо подготовить следующую задачу для симулятора:

- 1.1. Робот начинает свое движение в правом нижнем углу полигона с неизвестной ориентацией. На полигоне имеются четыре тумбы различных цветов. Цвет тумбы, наименее удаленной от стены полигона, задаёт направления координат и соответственно начальные координаты. На жёлтой тумбе с неизвестной стороны располагается `atTag` метка, кодирующая координаты финального сектора. Тумбы располагаются параллельно стенкам и координаты их центра задаются следующим образом: $(0, 5i; 0, 5j)$, где i, j — целые числа (т.е. тумбы могут быть смещены на пол сектора). Роботу необходимо доехать до точки финиша.
2. Основной файл с управляющей программой для проверки решения должен называться `main.py` и находиться в корне репозитория.
3. Merge Request необходимо назвать следующим образом: “код команды_`day3`”. Например: “`irs202100_day3`”.
4. Максимальное время выполнения одной попытки для задачи — 5 минут.
5. Решение будет проверяться на 2 проверочных полигонах.
6. Баллы за решение подзадач этапа:
 - 6.1. Робот смог определить расстояние между стенкой и ближайшей тумбой в метрах с точностью до десятых и вывел его в консоль в формате: `distance X` — 5 баллов.
 - 6.2. Робот нашел ближайшую к стенке тумбу, остановился и вывел в консоль её цвет в формате: `color X`, где X — цвет (red, green, blue, yellow) — 10 баллов.
 - 6.3. Робот нашел жёлтую тумбу, смог определить сторону с маркером. Остановился в направлении данного маркера и вывел в консоль: `found marker` и через 10 сек продолжил движение — 5 баллов.
 - 6.4. Робот нашел жёлтую тумбу, распознал маркер и вывел значения маркера в консоль в формате: `marker X Y` — 15 баллов.
 - 6.5. Робот определил свои глобальные координаты, находясь возле жёлтой тумбы, остановился, вывел координаты в консоль в формате: `coordinates X Y` и через 10 сек продолжил движение — 18 баллов.
 - 6.6. Проекция робота частично оказалась в секторе финиша, робот остановился и вывел в консоль `finish` — 12 баллов.
 - 6.7. Проекция робота полностью оказалась в секторе финиша, робот остановился и вывел в консоль `finish` — 5 баллов.
7. Значение координат необходимо вычислять для центра робота.
8. Значение координат необходимо вывести с точностью до целых.
9. Баллы за все попытки в каждой подзадаче суммируются.
10. Выполнение всех критериев в каждой из двух попыток всех подзадач дает дополнительные 10 баллов.
11. Максимальное количество баллов за этап — 150.

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.