|  |  |
| --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  Федеральное государственное бюджетное образовательное учреждение высшего образования  МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ) | |
|  |  |
|  |
|  |  |
| ОТЧЁТ  по дисциплине:  «АРХИТЕКТУРА ИНФОРМАЦИОННЫХ СИСТЕМ» | |
|  |
|  |  |
| Руководитель  Максимов А.Н.  « »   2023 года | Выполнили студенты группы М3О-221Б  Сунгатуллин С. Р.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шульц Б. Е.  « »   2023 года |
|  | |

Москва 2023

Оглавление

# ФОРМУЛИРОВКА ЗАДАНИЯ

Разработать ПО для робота, осуществляющего доставку напитков в заданном пространстве. ПО включает в себя:

1. Поиск заданных объектов на кадрах, полученных с камеры;
2. Передвижение робота по заданным командам;
3. Коммуникацию с пользователем посредством графического интерфейса;

# РЕАЛИЗАЦИЯ ОБРАБОТКИ ИЗОБРАЖЕНИЯ

Обработка изображения реализована посредством заголовочного файла ImageProcessing.h и файла ImageProcessing.cpp.

ImageProcessing.h содержит в себе два класса: Color и Detector.

Класс Color необходим для хранения цвета объектов в HSV формате.

Класс Detector включает в себя методы, необходимые для передачи информации о положении объектов заданного цвета на изображении и информации, необходимой для формирования соответствующей команды для передвижения робота.

Метод TakeThresholdOfBlob получает в качестве аргументов кадр с камеры и цвет. На выходе, метод формирует черно-белое изображение, выделяя на ней белым цветом объекты, попадающие в диапазон заданного цвета.

Метод detectBlob получает на вход исходный кадр с камеры и черно-белое изображение, сформированное методом TakeThresholdOfBlob. Данный метод находит наибольшее белое поле, помечает поле прямоугольником и возвращает координаты прямоугольника.

Метод GetMassCenter имеет реализацию для прямоугольника и для двух точек, подающихся в качестве аргументов. GetMassCenter, реализованный для прямоугольника, отбирает левый верхний и правый нижний точки входного прямоугольника и определяет координаты центра отрезка, образуемого двумя точками. GetMassCenter, реализованный для двух точек возвращает координаты центра отрезка, ограниченного данными двумя точками.

Метод findDistanceToDestination возвращает расстояние между двумя заданными точками.

Метод findAngle получает на вход координаты трех точек и возвращает косинус угла, между третьей точкой и прямой, проходящей через первые две точки.

Метод SteppedDetection получает на вход кадр с камеры и заданный цвет. На выходе, посредством методов TakeThresholdOfBlob, detectBlob и GetMassCenter метод возвращает центр масс объекта заданного цвета.

# РЕАЛИЗАЦИЯ ОБРАБОТКИ ЗАПРОСА И УПРАВЛЕНИЯ РОБОТОМ.

Обработка запроса и формирование команд управления роботом реализованы в заголовочном файле RequestProcessing.h и файле RequestProcessing.cpp.

RequestProcessing.h содержит в себе классы Request, Controller, и структуру AlphaBot.

Класс Request хранит в себе элементы запроса – цвет объекта и его координаты.

Метод SetColor получает на вход цвета объекта в текущем запросе и передаёт его на экземпляр класса Request.

Метод SetDestination получает на вход координаты местоположения объекта и передаёт их на экземпляр класса Request.

Метод SetType передаёт экземпляру класса Request тип подаваемого запроса.

Метод GetColor возвращает цвет объекта, хранящийся в экземпляре объекта Request.

Метод GetDestination возвращает координаты места назначения объекта, хранящийся в экземпляре объекта Request.

Метод GetType возвращает тип запроса, хранящийся в экземпляре объекта Request.

Класс Controller создаёт запрос, подаёт команды на робота для выполнения запроса, исходя из информации о положении места назначения и робота, и завершает запрос.

Метод MakeRequest получает в качестве аргументов цвет объекта и его координаты и подаёт входные данные на экземпляр класса Request.

Метод FinishRequest завершает выполнения запроса.

Метод Move отправляет роботу команду на движение вперёд.

Метод Rotate отправляет роботу команду на поворот вправо.

Метод GoHome имитирует отправление робота на станцию подзарядки.

Метод FiniteAutomate реализует конечный автомат состояний робота:

Состояние Running проверяет наличие запросов и, если запрос есть, меняет состояние робота на DoDeliver. Иначе, вызывает метод GoHome.

Состояние Disabling отключает доставку робота.

Состояние DoDeliver выбирает текущий кадр с камеры, обрабатывает его и выбирает следующее состояние, в зависимости от положения робота.

Состояние Move выполняет метод Move и переходит в состояние DoDeliver.

Состояние Rotate выполняет метод Rotate и переходит в состояние DoDeliver.

Состояние Waiting имитирует состояние ожидания чего-либо (ожидание кружки на роботе) и переключает состояние в Running.

Структура AlphaBot хранит координаты «головы» и «хвоста» робота на кадре.

# РЕАЛИЗАЦИЯ ПЕРЕСЫЛКИ СООБЩЕНИЙ

Межпроцессное взаимодействие реализовано в заголовочном файле Mqtt.h и файле Mqtt.cpp.

Mqtt.h содержит в себе класс myMosq и структуру MqttMessage.

Класс myMosq реализует получение нужного запроса из топика и отправку выбранной команды на топик робота.

Методы on\_connect, on\_subscribe и on\_message – вспомогательные методы обработки состояний для экземпляра класса myMosq. Получение сообщения осуществляется в методе on\_message.

Метод Subscribe создает цикл для обмена сообщениями в топике.

Метод SendToServer осуществляет отправку сообщения в заданный топик.

Метод Publish создаёт объект типа json в соответствии с полученной командой и отправляет его в заданный топик.

Структура MqttMessage хранит информацию о команде, которую нужно отправить роботу.

# РЕАЛИЗАЦИЯ Коммуникации с пользователем посредством графического интерфейса

Коммуникация с пользователем посредством графического интерфейса реализована через telegram бота.

Бот выдает пользователю список доступных цветов объекта на выбор и ждёт, пока пользователь сделает выбор. После получения ответа от пользователя, бот передаёт цвет объекта на сервер.