

Регламент проведения квалификации для хакатона  
“Умный беспилотный автомобиль СтарЛайн”  
2020 г.

Контактная информация:

Организационный комитет:  
Почта: **hackathon@robofinist.ru**

Технические вопросы: **Дема Николай**  
Тел: +7 (981) 681-06-52  
Почта: **dema.nu@starline.ru**

## Содержание

1.	Общие положения	2
2.	Описание предоставляемого участникам программного обеспечения	3
2.1.	Подготовка к работе	3
2.2.	Работа с docker-контейнером	4
2.3.	Работа с симулятором и организация процесса разработки	5
3.	Описание квалификационного задания	7
3.1.	Порядок представления решения	8
3.2.	Допускаемые изменения предоставляемого программного обеспечения	9
4.	Оценка выполнения задания	10

## **1. Общие положения**

Целью проведения квалификации является отбор команд для непосредственного участия в хакатоне “Умный беспилотный автомобиль СтарЛайн”.

Квалификационное задание призвано помочь организаторам хакатона предварительно оценить не только компетенции команды в технических областях знаний, но возможности к поиску неординарного, творческого подхода к решению задач в короткие сроки, а также полноту и ясность в представлении результатов выполнения задания.

За подробной информацией по вопросам организации и порядке участия в хакатоне “Умный беспилотный автомобиль СтарЛайн” пожалуйста обращайтесь к [официальной странице мероприятия](#).

## 2. Описание предоставляемого участникам программного обеспечения

Все необходимое для выполнения квалификационного задания программное обеспечение доступно в [официальном репозитории хакатона](#).

Участникам предоставляется симуляция небольшого города и модели turtlebot 2 в нем. В качестве симулятора используется [gazebo](#). В качестве основного фреймворка для разработки используется [ROS](#). Работа с симулятором предполагается внутри docker-контейнера, скрипты и утилиты для сборки которого также предоставляются в составе репозитория.

Тестирование предоставляемого программного обеспечения проводилось на операционной системе Ubuntu версий 16.04 и 18.04. Соответственно, все инструкции, описываемые в данном регламенте, предполагают использование одной из этих систем и мы крайне рекомендуем использовать именно их.

В случае наличия проблем с предоставляемым программным обеспечением вы можете обратиться за помощью используя [контактную информацию](#).

### 2.1. Подготовка к работе

Для сдачи задания вам потребуется создать форк [репозитория хакатона](#) на [gitlab](#). Поэтому прежде всего заведите командный аккаунт, если такового еще нет, и создайте соответствующий форк.

Далее склонируйте ваш форк и перейдите в корневую директорию репозитория:

```
git clone https://gitlab.com/your_repo/hackathon_kobuki.git
cd hackathon_kobuki
```

Затем установите Docker. Для этого можете воспользоваться скриптом:

```
bash scripts/docker_install.bash
```

В случае, если на вашем ПК используется видеокарта от nvidia, то обозначенный выше скрипт следует исполнить с параметром -n или --nvidia.

После этой операции следует перезайти в систему для корректной работы docker.

Для установки docker вручную воспользуйтесь [официальной инструкцией](#). В случае, если на вашем ПК используется видеокарта от nvidia, то требуется дополнительно установить [nvidia container toolkit](#).

## 2.2. Работа с docker-контейнером

Для работы с докер-контейнером доступны следующие скрипты:

```
docker/simulator/build_docker.sh - сборка контейнера  
docker/simulator/run_docker.sh - запуск контейнера  
docker/simulator/into_docker.sh - запуск bash в контейнере
```

Соответственно, соберите и запустите контейнер выполнив последовательно:

```
bash docker/simulator/build_docker.sh  
bash docker/simulator/run_docker.sh
```

В случае, если на вашем ПК используется видеокарта от nvidia, то обозначенные выше скрипты следует исполнить с параметром -n или --nvidia.

## 2.3. Работа с симулятором и организация процесса разработки

В корне репозитория располагаются два [catkin\\_tools](#) окружения: **simulator\_ws** и **catkin\_ws**. В **simulator\_ws** располагаются модели, файлы описания и плагины, требующиеся для симуляции **turtlebot**, **catkin\_ws** - пользовательское окружение, созданное специально для размещения участниками решения квалификационного задания. Оба окружения монтируются внутрь контейнера, таким образом разработку можно вести как внутри, так и вне контейнера используя вашу любимую IDE.

При первом входе в контейнер требуется собрать пакеты в окружении **simulator\_ws**, для этого в терминале, в котором вы выполнили предыдущие две команды, выполните:

```
cd /simulator_ws
catkin build
```

Далее попробуйте запустить симуляцию используя **launch**-файл из пакета **tb\_gazebo**:

```
roslaunch tb_gazebo turtletown.launch
```

После этого в появившемся окне **gazebo** вы должны увидеть примерно следующую картинку:



При попытке запуска обозначенной выше команды после первой сборки окружения `simulator_ws` в том же терминале вы получите ошибку, так как для текущей `bash`-сессии переменная окружения `PATH` еще не содержит соответствующих путей до используемого пакета, для того, чтобы исправить это выполните:

```
source /simulator_ws/devel/setup.bash
```

Далее запустите новый терминал, откройте еще одну `bash`-сессию в контейнере и запустите `teleop`:

```
bash <path-to-your-rep>/docker/simulator/into_docker.sh  
roslaunch tb_gazebo teleop.launch
```

В случае, если вы можете управлять роботом, то подготовку к работе с симулятором можно считать законченной.

### 3. Описание квалификационного задания

В рамках квалификационного задания участникам требуется решить комплексную задачу автономной навигации в заранее неизвестном пространстве.

В качестве оперативного пространства выступает модель города со статическими препятствиями. Мобильный робот предоставляет информацию об одометрии и возможных столкновениях передним бампером. Робот также оснащен rgb-d-камерой и imu. Все эта информация доступна в соответствующих gos-топиках.





Для прохождения квалификации робот из точки старта (область **S** на рисунке выше, обозначена зеленым) должен автономно достичь свободного парковочного места перед магазином (область **F**, обозначена синим) с предотвращением столкновения с препятствиями.

Выбор подхода и программных средств к решению поставленной задачи в рамках проведения квалификации не ограничивается.

### 3.1. Порядок представления решения

В качестве решения квалификационного задания команда-участник предоставляет организаторам хакатона ссылку на публичный форк [официального репозитория хакатона](#) на [gitlab](#), в котором будет содержаться программное обеспечение для решения квалификационного задания и инструкции по его использованию. Также команда-участник должна предоставить видео, демонстрирующее работу представляемого решения.

Необходимые ссылки, а также любые дополнительные материалы следует предоставить организационному комитету в виде письма по адресу [hackathon@robofinist.ru](mailto:hackathon@robofinist.ru). В теме письма укажите название вашей команды, затем “Решение квалификационного задания” без кавычек.

Со сроками выполнения квалификационного задания можно ознакомиться на [официальной странице мероприятия](#).

## 3.2. Допускаемые изменения предоставляемого программного обеспечения

- 3.2.1. Допускаются любые правки предоставляемого docker-контейнера, например включения в его состав зависимостей для предоставляемого решения.
- 3.2.2. Допускаются изменения состава сенсорики мобильного робота, например добавление дополнительных камер или лидара.
- 3.2.3. Не допускается изменять статичную модель города (mesh и текстуры).
- 3.2.4. Не допускается использовать дополнительную сенсорику не входящую в состав робота (активная инфраструктура).
- 3.2.5. Не допускается изменять динамические параметры модели робота, если этого не требует пункт 3.2.2.
- 3.2.6. Разрешается добавлять динамические препятствия в модель города, например для демонстрации возможностей предоставляемого решения.

В случае любых правок со стороны участников требуется обеспечить возможность тестирования предоставляемого решения путем предоставления соответствующих инструкций в составе README форка.

#### 4. Оценка выполнения задания

Оценка квалификационного задания несет скорее качественный, нежели количественный характер. Особое внимание при оценке будет уделяться конечному составу используемой в составе робота сенсорики, робастности решения к условиям окружения и его требовательности к вычислительным ресурсам.

Положительным результатом оценки является прием заявки команды на участие в хакатоне. Информация о приеме заявок отображается на сайте РобоФинист и поступает на электронную почту участникам.

В случае публичной работы над решением квалификационного задания существует возможность того, что одна команда может использовать наработки другой. Для предотвращения таких ситуаций до окончания сроков проведения квалификации рекомендуется использовать командный репозиторий в приватном режиме.

В случае возникновения неоднозначных ситуаций касательно порядка проведения квалификации или условий выполнения задания, не регламентированных данным документом, свяжитесь с нами, используя [контактную информацию](#).