

Регламент проведения заключительного этапа

online



Innopolis Open

Innopolis University Olympiad
in Robotics

2020

профиль

**МАНИПУЛЯЦИОННЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ
РОБОТОТЕХНИЧЕСКИЕ СИСТЕМЫ**

Содержание

1. Площадка проведения состязаний.....	3
2. Расписание финального этапа.....	3
3. Дополнительные задания.....	3
4. Порядок выполнения задания	4
4.1. Проверки и тренировки	4
4.2. Ограничения на использование команд.....	4
4.3. Отправка решений.....	5
4.4. Попытки.....	6
5. Порядок проверки решений.....	6
6. Составление итогового рейтинга команд и определение победителей	7

1. Площадка проведения состязаний

- 1.1. Финал олимпиады Innopolis Open Robotics по профилю «Манипуляционные интеллектуальные робототехнические системы» (далее – МаниРС) проводится в дистанционном режиме (онлайн).
- 1.2. Для проведения соревнований в обеих возрастных группах ([6-8 классы](#) и [9-11 классы](#)) применяется симулятор роботов CoppeliaSim (Education версия для бесплатного использования в образовательных задачах). На страницах профилей сайта олимпиады выкладываются архивы с инструментарием для участников (сценами для симулятора, шаблонами программ и т.д.).

2. Расписание финального этапа

- 2.1. Финальный этап проводится в течение трех дней: с 19 по 21 июня. Программа следующая:
 - 1) 19 июня – тренировочный день;
 - 2) 20 июня – первый зачетный день;
 - 3) 21 июня – второй зачетный день.
- 2.2. В тренировочный день командам предлагается решить стандартное задание олимпиады (описанное на страницах соответствующих возрастных групп). В этот день тестируется рассылка заданий и инструментария, решение заданий участниками на локальных компьютерах, отправка решений организаторам и их проверка судьями. Время проведения тренировки: с 9:00 МСК до 15:00 МСК.
- 2.3. В зачетные дни командам предлагается решить не только стандартное задание олимпиады, но и дополнительные. В каждый зачетный день будет предложено отдельное дополнительное задание. Расписание каждого зачетного дня представлено в таблице 1.

Часовой пояс	KALT	MSK	SAMT	YEKT	OMSK	KRAT	IRKT	YAKT	VLAT
Оглашение дополнительного задания	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00
Попытка 1	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00
Попытка 2	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00
Попытка 3	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00

Table 1. расписание зачетных дней для разных часовых поясов

3. Дополнительные задания

- 3.1. Оглашение дополнительного задания происходит одновременно для всех участников, в 9:00 по московскому времени в соответствующий зачетный день.

3.2. Задание отправляется тренеру команды на электронную почту, указанную при регистрации, в виде архива.

3.3. Архив содержит следующие элементы:

- 1) Текстовое описание задания с описанием дополнительного задания, выполненного согласно [“Правил формирования дополнительных заданий”](#).
- 2) Сцену для симулятора CoppeliaSim с полигоном, игровыми элементами, стандартным роботом; в том числе содержащую элементы дополнительного задания.
- 3) Шаблоны программ на языках программирования Python, C++, Lua.

4. Порядок выполнения задания

4.1. Проверки и тренировки

- 4.1.1. Участники скачивают архивы, знакомятся с заданием и готовят программу для его выполнения моделью робота.
- 4.1.2. Тренировки проводятся участниками локально на собственных компьютерах.
- 4.1.3. Изменять физические параметры полигона, игровых объектов и узлов робота запрещено (зачетный запуск программы будет производиться судьями на исходной сцене симулятора).
- 4.1.4. Разрешено изменять скорости перемещения звеньев и коэффициенты их ПИД-регуляторов, а также параметры симуляции, такие как точность выполнения симуляции, длительность шага симуляции, симуляция в реальном времени и т.д.
- 4.1.5. Допускается создание дополнительных скриптов для узлов и элементов робота (его камеры, схвата или какого-либо звена).
- 4.1.6. Создание дополнительных скриптов производится участниками на свой страх и риск изменения (ухудшения работы) модели робота.
- 4.1.7. Все эти изменения должны быть описаны простым и понятным языком в дополнительном файле Changes.pdf.
- 4.1.8. Организаторы попытаются перенести описанные в файле изменения в исходную модель перед запуском зачетной попытки.

4.2. Ограничения на использование команд

- 4.2.1. При подготовке решения допускается использовать сторонние библиотеки команд для языков программирования Python и C++ (например, для обработки изображений). В тренировочный день тренер команды должен уведомить организаторов об используемых библиотеках и предоставить инструкции для их установки и настройки. Организаторы постараются установить их на систему, используемую при проверке решений, но не гарантируют их полную работоспособность. Поэтому рекомендуется реализовывать ключевые функции самостоятельно.



- 4.2.2. При подготовке решения запрещается использовать API-команды симулятора, мгновенно перемещающие игровые элементы или детали робота, устанавливающие жесткие связи между объектами или изменяющие / отключающие их динамические свойства. Например, запрещено использовать команду `setJointPosition()` вместо `setJointTargetPosition()`: первая мгновенно устанавливает звено робота в указанное положение, при использовании второй в указанное положение манипулятор приводится встроенным ПИД-регулятором.
- 4.2.3. Кроме того, запрещено использовать API-команды симулятора для определения свойств и расположения игровых элементов. Организаторы постарались подготовить необходимые для выполнения задания функции, описанные в файле `ReadMe.pdf` каждой возрастной группы. Рекомендуется использовать их. Если участникам недостаточно этих функции и есть предложения по созданию дополнительных, то они должны быть озвучены до 10:00 МСК 19 июня (тренировочного дня) на официальном форуме профиля.
- 4.2.4. Основным критерием определения запрещенных команд является отсутствие их аналогов у реального робота: физический робот может узнать информацию об окружающем мире только посредством датчиков, перемещать игровые объекты с помощью механического перемещения. Не допускается использование сторонних библиотек и команд, определяющих положения звеньев робота автоматически перемещающих их из одной точки в другую, так как именно эту задачу участники должны решить самостоятельно.
- 4.2.5. Решения, использующие запрещенные команды, не будут учитываться в результатах команды, их результат будет 0 баллов и 120 секунд за попытку.

4.3. Отправка решений

- 4.3.1. Тренер команды присылает организаторам архив с решением или загружает его в облачное хранилище или репозиторий, предоставленные организаторами. Окончательное решение об используемой системе отправки решений будет озвучено в тренировочный день 19 июня. Решение (архив) должно содержать следующие элементы:
- 1) Сцену для симулятора с настроенными первичными параметрами, разрешенными для редактирования (разрешается настроить скорости и коэффициенты ПИД-регуляторов звеньев в сцене, а не через программу и т.д.)
 - 2) Файл с исходным кодом программы. Программа на Python должна быть сохранена в файле с расширением `.py`, на Lua – в файлы `Child.lua` (код из потокового скрипта сцены) и `Customization.lua` (код из кастомизирующего скрипта сцены), на C++ – с расширением `.cpp`. Именно код из этих файлов будет запускаться при проверке в автоматическом режиме.



- 3) Дополнительные файлы с исходным кодом вспомогательных функции, вызываемых в основной программе. Они могут располагаться как в основном каталоге архива (рядом с файлом основной программы), так и соблюдая какую-либо иерархию, выбранную самими участниками и описанную при вызове из основной программы. Организаторы рекомендуют соблюдать иерархию архива с заданием
- 4) Файл Changes.pdf с описанием изменений скриптов сцены. Наличие этого файла сигнализирует организаторам о том, что в исходную сцену надо вручную перенести какие-либо скрипты или параметры. О том, какие параметры переносятся в автоматическом режиме, смотрите в пункте 5.4.

4.3.2. Команды могут отправлять или обновлять решения в репозиториях неограниченно количество раз перед попыткой.

4.4. Попытки

- 4.4.1. В определенные моменты времени, обозначенные в таблице 1, текущие версии решений будут отправляться на проверку. Результаты проверки идут в зачет как Попытка 1...3 соответствующего дня.
- 4.4.2. Сохранение текущих версий решений происходит автоматически по таймеру, так что рекомендуем учитывать время на загрузку или синхронизацию облачных хранилищ и отправлять решение за несколько минут до обозначенного в таблице 1 времени.

5. Порядок проверки решений

- 5.1. Для проведения попытки решение от участников запускается в автоматическом режиме 5 раз. В зачет идет результат запуска с лучшими баллами. При наличии нескольких запусков с одинаковыми баллами, в зачет идет тот из них, который занял меньшее время.
- 5.2. В каждой попытке программы всех участников запускаются в сцене с одинаковой стартовой расстановкой игровых объектов.
- 5.3. Для автоматической проверки принимаются последние версии решений, даже если они были загружены еще во время подготовки к предыдущей попытке. Например, команда может прислать одно решение в 11:00 МСК и это решение будет учтено для всех трех попыток этого зачетного дня.
- 5.4. Проверка происходит в автоматическом режиме в следующем порядке:
 - 1) В исходную сцену специальной программой переносятся стартовые параметры сцену участников. Конкретно переносятся следующие параметры: максимальные скорости каждого привода робота, коэффициенты их ПИД-регуляторов, их стартовые положения, параметры симуляции.
 - 2) К симулятору подключается программа-судья, выполняющая первичную расстановку игровых объектов и определение нулевого положения робота.



- 3) Запускается основная программа участника. Программа на C++ открывается, компилируется и запускается в IDE Visual Studio 2019. Программа на Lua не копируется в сцену, а запускается из файла с исходным кодом, содержащимся в архиве.
- 4) Программа-судья на каждом шаге симуляции проверяет некоторые параметры робота, такие как максимальная скорость приводов или динамические параметры. Если параметры превышают допустимые, то программа-судья возвращает их к допустимым значениям.
- 5) При окончании попытки или возникновении ошибки в программе участника, программа-судья фиксирует результат выполнения попытки (расположение объектов, время остановки по симулятору, возвращение звеньев в нулевое положение).
- 6) Если в архиве содержится файл Changes.pdf (ровно с таким именем, проверяется автоматически), то перед запуском симуляции организаторы в ручном режиме переносят дополнительные скрипты из сцены участников в исходную сцену, на которой происходит проверка.

6. Составление итогового рейтинга команд и определение победителей

6.1. У каждой команды фиксируются результаты баллов и времени выполнения шести попыток: по три в каждый зачетный день. Они ранжируются следующим образом:

День 1		День 2	
Лучшая попытка	Попытка с максимальными баллами. Если таких несколько – то с наименьшим временем выполнения	Лучшая попытка	Попытка с максимальными баллами. Если таких несколько – то с наименьшим временем выполнения
Средняя попытка	Попытка, из оставшихся кроме лучшей, с наибольшими баллами. Если таких несколько – то с наименьшим временем выполнения	Средняя попытка	Попытка, из оставшихся кроме лучшей, с наибольшими баллами. Если таких несколько – то с наименьшим временем выполнения
Худшая попытка	Оставшаяся попытка	Худшая попытка	Оставшаяся попытка

6.2. Общий рейтинг участников выстраивается с учетом следующих критериев (в порядке их уменьшения):

- 1) сумма баллов лучших попыток двух дней – в топе рейтинга участники с большим количеством баллов;
- 2) сумма баллов средних попыток – при равенстве суммы баллов лучших попыток, в топ рейтинга попадают участники с большей суммой баллов средних попыток;



- 3) сумма баллов худших попыток – при равенстве предыдущих пунктов, в топ выходят участники с максимальной суммой баллов худших попыток;
- 4) сумма времени лучших попыток – при равенстве предыдущих пунктов, в топ выходят участники с минимальной суммой времени лучших попыток;
- 5) сумма времени средних попыток – при равенстве предыдущих пунктов, в топ выходят участники с минимальной суммой времени средних попыток;
- 6) сумма времени худших попыток – при равенстве предыдущих пунктов, в топ выходят участники с минимальной суммой времени худших попыток.