

Командный практический тур

Описание задачи финала

Целью командного этапа является выявление команды, которая найдет лучшее решение автоматизации процесса строительства. Участникам предстоит разработать и запустить цикл постройки здания, используя машинное зрение и промышленные манипуляторы.



Задача

Участникам предстоит разработать и запустить цикл постройки здания. При разработке необходимо использовать машинное зрение, а также пользоваться принципами работы промышленных манипуляторов Kawasaki. Решение должно учитывать доступные технологии: 3D-печать, симуляция производственного цикла, машинное зрение, удаленный доступ. Конструкцию манипулятора необходимо доработать таким образом, чтобы обеспечить безопасность автоматизированной постройки здания.

Готовность на момент начала финала

На момент начала финала участники должны иметь:

- знания в среде программирования и симуляции манипуляторов Kawasaki: K-ROSET, KIDE (Ссылка на вебинар);
- умение работать с библиотекой обработки изображения, умение развернуть эту систему на новом компьютере;
- САПР на своем устройстве, умение проводить прочностной расчет.

Подзадачи финала

Командные баллы складываются из трех разделов, оцениваемых по 18, 27 и 55 баллов. Максимальное количество баллов составляет 100.

Задача III.2.0.1. Безопасная проставка (18 баллов)

Цель задания

Необходимо изготовить проставку, соединяющую электромагнит и манипулятор. Проставка защищает манипулятор от поломки в случае ошибки в коде. Это задание также будет использоваться как допуск для работы над заданиями 4.5.9–4.5.11.



Участникам необходимо сделать проставку, срабатывающую как можно ближе к нагрузке в 50 Н. Срабатыванием считается защита робота от поломки при уведении фланца робота на высоту -10 см от рабочей плоскости. Данное уведение в целях безопасности имитируется приложением вертикальной нагрузки на проставку и измерением количества сжатия проставки с нагрузкой на фланец.

Система оценки

- Близость при срабатывании к эталонной нагрузке — 50 Н.
- Многообразие проставки.
- Простота решения (возможность серийного производства).

Решение

Пример хорошо сделанной проставки, получившей максимальный балл за попадание в нагрузку, многообразие и простоту.



Задача III.2.0.2. OpenCV (15 баллов)

Цель задания

Участникам необходимо написать программу для распознавания объемных фигур (блоков) на полигоне. Решение данной задачи в дальнейшем будет использоваться для всех задач блока 3.2.3.

Задача программы

- Найти координаты всех блоков, расположенных на полигоне.
- Определить тип блока.
- Определить градус поворота фигуры
- В качестве ответа выдать файл .json с указанными выше параметрами.

Решение

Машинное зрение участниками было написано в том числе при помощи библиотеки OpenCV для языка программирования Python3. Также в решениях было использовано распознавание ArUco-маркеров, расположенных на полигоне.

Система оценки

Баллы участникам выставлялись в соответствии с рейтинговой системой, основанной на сумме отклонений между реальными и распознанными участниками координатами фигуры.

Задача III.2.0.3. K-ROSET (12 баллов)

Цель задания

Продемонстрировать навыки программирования промышленного манипулятора Kawasaki в виртуальной среде для получения допуска к работе на реальном полигоне.

Задача программы

Участникам необходимо написать программу на языке программирования AS для постройки дома из блоков. Участникам была предоставлена модель полигона в формате .stl, на котором в случайном порядке были расставлены блоки, а также фотография необходимой финальной постройки. Программы запускались в симуляторе K-ROSET — программном обеспечении, разработанном специально для моделирования автономных роботов Kawasaki.

Решение

Ниже приведены пример части управляющей программы, а также скриншоты из симулятора K-ROSET до выполнения программного кода и после.

```

1  .PROGRAM main()
2  SPEED 300 MM/S ALWAYS
3  ACCURACY 5
4  OPENI
5  HOME
6  ;
7  JMOVE SHIFT(arch_take_c_put BY ,,110)
8  LMOVE arch_take_c_put
9  CLOSEI
10 LMOVE SHIFT(arch_take_c_put BY ,,110)
11 ;
12 JMOVE SHIFT(arch_put BY ,,110)
13 LMOVE arch_put
14 OPENI
15 LMOVE SHIFT(arch_put BY ,,110)
16 ;
17 JMOVE SHIFT(cube_take BY ,,110)
18 LMOVE cube_take
19 CLOSEI
20 LMOVE SHIFT(cube_take BY ,,200)
21 ;
22 JMOVE SHIFT(cube_put BY ,,200)
23 LMOVE cube_put
24 OPENI
25 LMOVE SHIFT(cube_put BY ,,200)
26 ;

```

```
27 ;
28 ;
29 JMOVE SHIFT(cone_take BY ,,110)
30 LMOVE cone_take
31 CLOSEI
32 LMOVE SHIFT(cone_take BY ,,110)
33 ;
34 JMOVE SHIFT(cone_put BY ,,110)
35 LMOVE cone_put
36 OPENI
37 LMOVE SHIFT(cone_put BY ,,110)
38 ;
39 ;
40 ;
41 JMOVE SHIFT(p_take BY ,,150)
42 LMOVE p_take
43 CLOSEI
44 LMOVE SHIFT(p_take BY ,,220)
45 ;
46 JMOVE SHIFT(p_put BY ,,220)
47 LMOVE p_put
48 OPENI
49 LMOVE SHIFT(p_put BY ,,230)
50 ;
51 ;
52 ;
53 JMOVE SHIFT(cylinder_take BY ,,200)
54 LMOVE cylinder_take
55 CLOSEI
56 LMOVE SHIFT(cylinder_take BY ,,200)
57 ;
58 JMOVE SHIFT(cylinder_put BY ,,200)
59 LMOVE cylinder_put
60 OPENI
61 LMOVE SHIFT(cylinder_put BY ,,200)
62 ;
63 HOME
64 .END
```

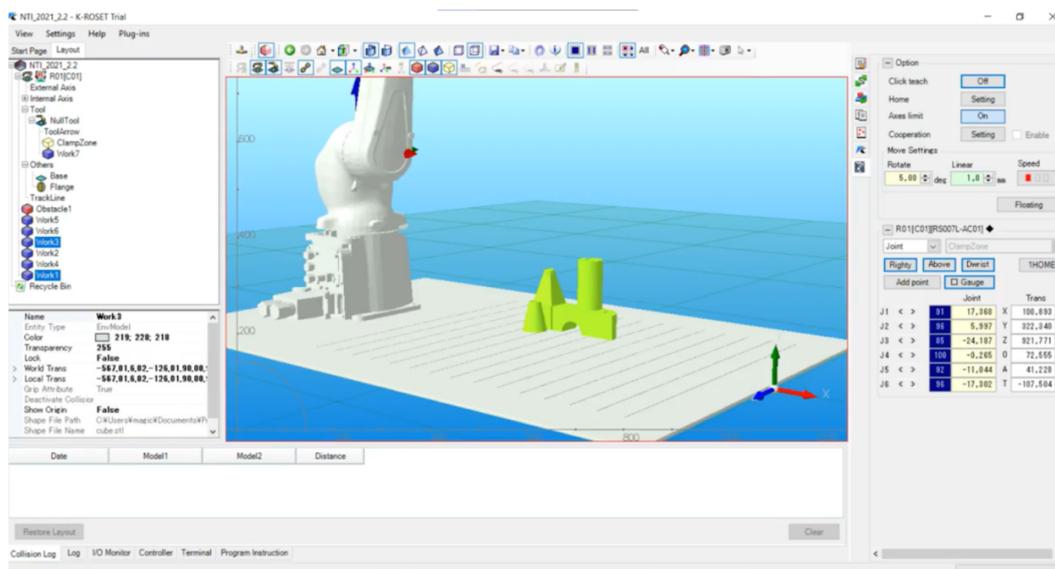


Рис. III.2.1: Расположение блоков до выполнения программы

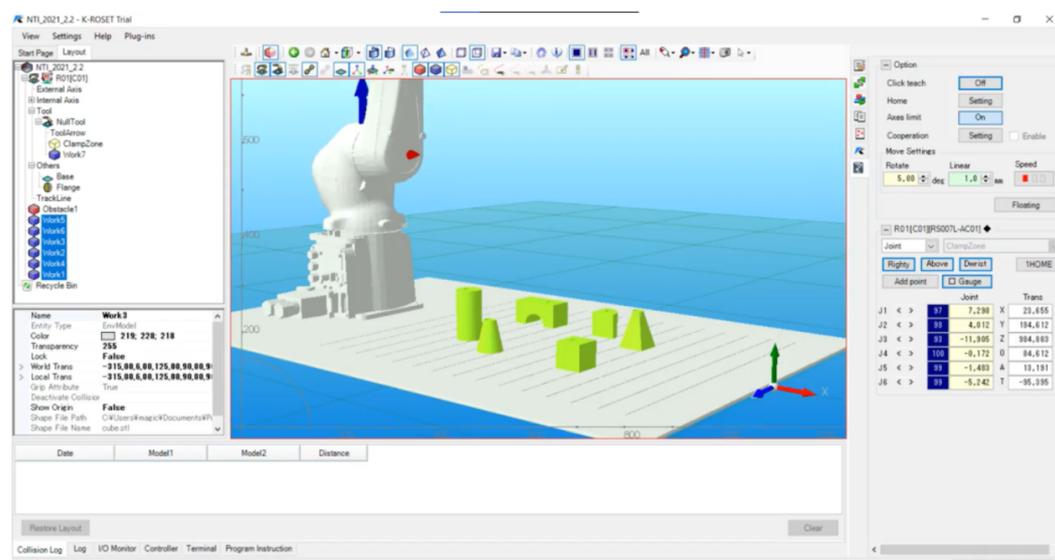


Рис. III.2.2: Расположение блоков после выполнения программы

Система оценки

Максимальный балл выставлялся участникам при соблюдении следующих условий:

- Перемещение блоков происходит на безопасной высоте в 15 мм
- Манипулятор не совершает лишние и/или хаотические движения
- Скорость перемещения манипулятора оптимальна для поставленной задачи
- После выполнения программы манипулятор вернулся в исходную точку

Для следующего блока заданий была определена следующая общая система штрафов:

- Падение блока с высоты большей 3 мм: -2 балла;

- Участники запустили выполнение программы на работе без разрешения экспертов: -2 балла и повторное прохождение испытания;
- Сломано крепление (проставка) электромагнита: -2 балла + повторное прохождение испытания;
- Перемещение происходит не на безопасной высоте — 10 сантиметров над блоками: -2 балла.

Задача III.2.0.4. Управление магнитным захватом (3 балла)

Цель задания

Проверка возможности дистанционного подключения участников к персональным компьютерам, расположенным на площадке и подключенным к манипуляторам. Проверка работоспособности кода, написанного участниками, на манипуляторах.

Задача программы

Участникам требовалось написать программу, которая в течение определенного количества времени выполняла циклический алгоритм, включающий в себя включение и выключение магнитного захвата, перемещение манипулятора по двум осям в двух направлениях, а также задержку.

Решение

Ниже предоставлен пример работающей программы, управляющей манипулятором следующим образом:

1. Захват выключен;
2. 3 секунды задержка;
3. Захват включен;
4. 3 секунды задержка;
5. Перемещение по оси X на +10 см;
6. Перемещение по оси Y на +10 см;
7. Перемещение по оси X на -10 см;
8. Перемещение по оси Y на -10 см;
9. 3 секунды задержка, а затем, по истечении 30 секунд манипулятор с выключенным захватом возвращается в первоначальное положение.

```

1 .PROGRAM main()
2 SPEED 100 MM/S ALWAYS
3 ACCURACY 0.5 ALWAYS
4 CP OFF
5 SINGULAR OFF
6 TIMER 1 = 0
7
8 DO
9   SIG -10
10  TWAIT 3
11  IF TIMER(1)>=30 THEN
12    HALT
13    END
14  SIG 10

```

```
15 TWAIT 3
16 IF TIMER(1)>=30 THEN
17     HALT
18     END
19 DRAW 100
20 IF TIMER(1)>=30 THEN
21     HALT
22     END
23 DRAW ,100
24 IF TIMER(1)>=30 THEN
25     HALT
26     END
27 DRAW -100
28 IF TIMER(1)>=30 THEN
29     HALT
30     END
31 DRAW ,-100
32 IF TIMER(1)>=30 THEN
33     HALT
34     END
35 TWAIT 3
36 UNTIL TIMER(1) >= 30
37 .END
```

Задача III.2.0.5. Перенести блок из точки A в точку B, координата A заранее неизвестна (6 баллов)

Цель задания

Проверка навыков участников, по объединению нескольких частей задачи: работой с машинным зрением и генерацией программного кода на языке AS для корректной работы манипулятора.

Задача программы

- Распознавание объекта и его расположения на полигоне при помощи компьютерного зрения.
- Чтение файла с расширением json с конечными координатами блока.
- Генерация кода для манипулятора, выполняющего перемещение блока из начального расположения в точку с указанными координатами.

Система оценки

Максимальный балл получили команды, выполнившие задачу.

Команды, неправильно определившие точку A и/или точку B на полигоне, получали штраф размером в 1 балл и отправлялись на повторное прохождение испытания.

Задача III.2.0.6. Постройка 1: Низкая сложность (12 баллов)

Данная подзадача, как и все последующие, служит проверкой навыков участников по объединению всех уже готовых модулей. Каждая из задач предусматривала похожую логику работы программы, однако с повышением уровня сложности добавлялись новые входные условия и убирались некоторые ограничения.

Задача программы

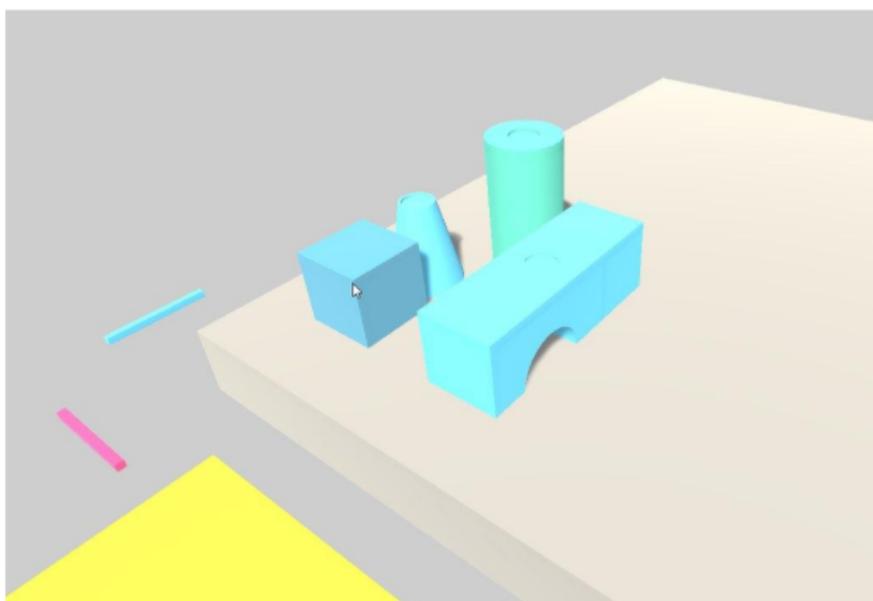
Участникам необходимо на основе фотографии полигона, на котором расположено несколько блоков, а также .json файла, который содержит в себе расположение всех фигур, сгенерировать программу, целью которой будет собрать постройку, описанную в .json файле, перенеся все блоки из их текущего положения на полигоне.

Ключевые ограничения и условия:

- Блоки расположены под прямым углом к линиям полигона
- Высота финальной постройки не превышает один блок
- Постройка состоит из 5 блоков

Решение

Пример постройки, которую необходимо было собрать участникам:



Задача III.2.0.7. Постройка 2: Средняя сложность (14 баллов)

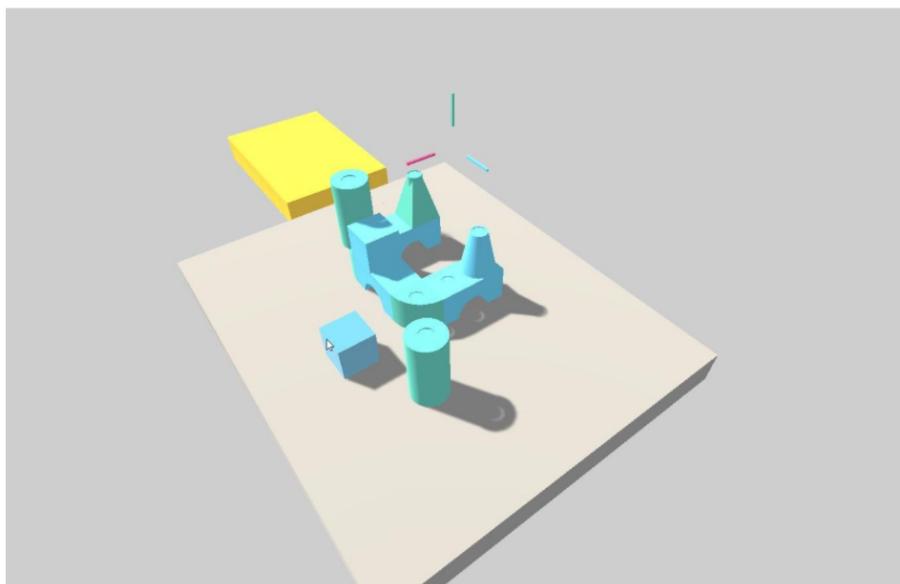
Данная подзадача является продолжением предыдущего задания.

Ключевые ограничения и условия:

- Блоки расположены под произвольным углом к линиям клеток полигона, но вектор нормали к металлической пластине всегда направлен вертикально вверх
- Максимальная высота финальной постройки 3 блока
- Постройка состоит из 11 блоков

Решение

Пример постройки, которую необходимо было собрать участникам:



Задача III.2.0.8. Постройка 3: Повышенная сложность (20 баллов)

Данная подзадача является продолжением предыдущих двух заданий.

Ключевые ограничения и условия:

- Блоков на полигоне в изначальной расстановке может быть больше, чем необходимо для постройки
- Максимальная высота финальной постройки 5 блоков
- Постройка состоит из 12 блоков

Решение

Пример постройки, которую необходимо было собрать участникам:

