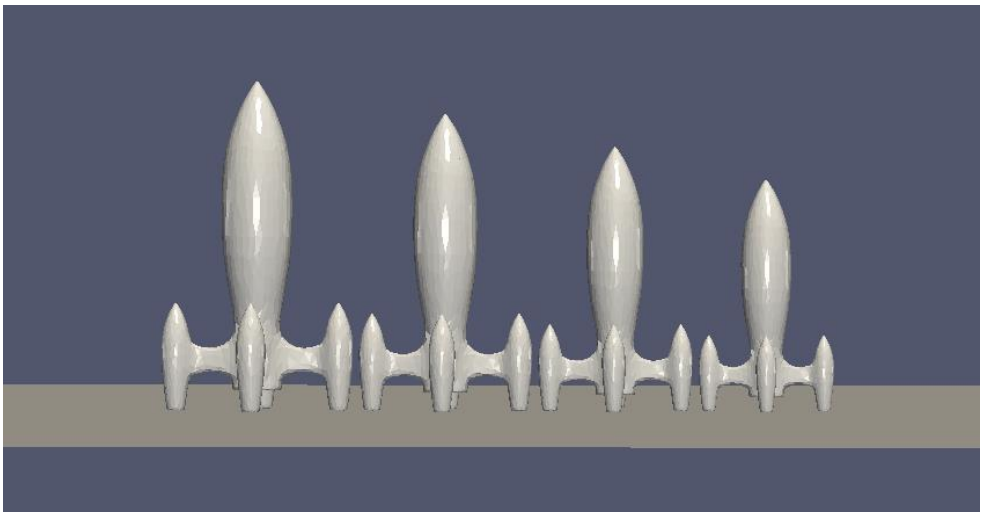
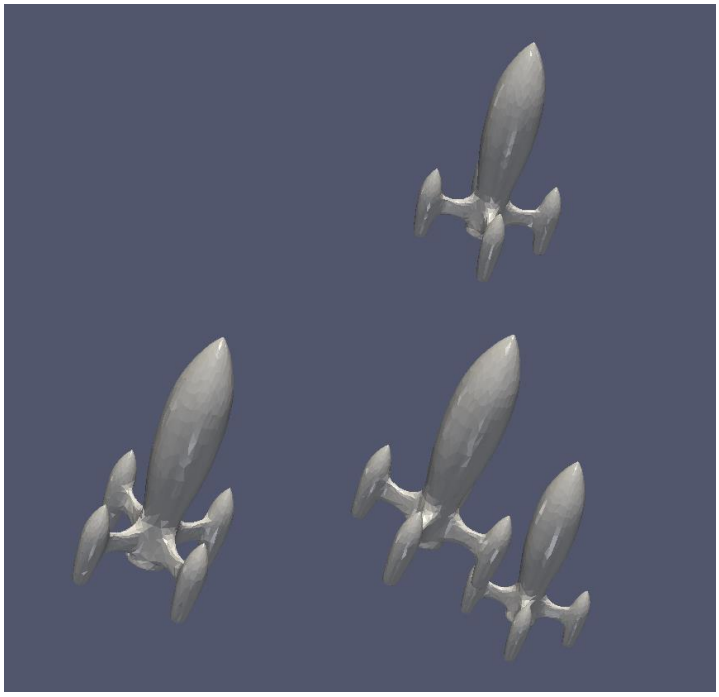
“Spaceships landing”

Описание

Есть несколько космических кораблей в пространстве космодрома. Задача диспетчера посадить эти корабли на космодром в одну линию, упорядочив их по размеру, использовав минимальное количество команд.



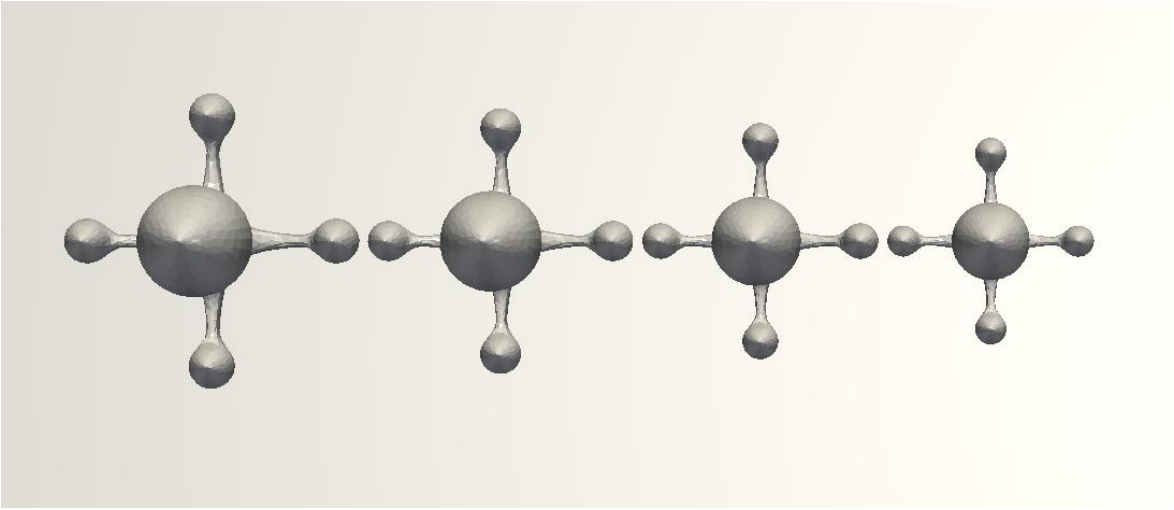


Детали и ограничения

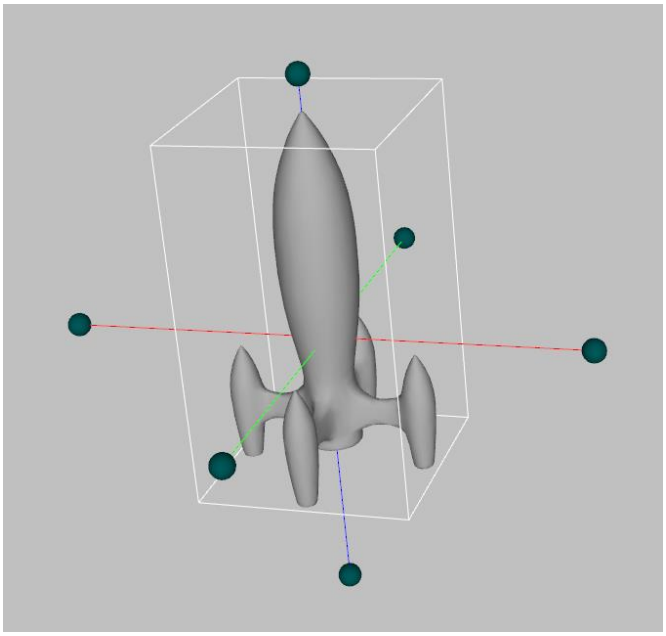
1. Все корабли имеет одну форму, отличаются лишь размером и получены путем линейного масштабирования исходной модели с коэффициентами 0.7, 0.8, 0.9, 1.0.

2. Число кораблей $0 < N \leq 10$.
3. Все линейные размеры приведены в одних единицах. Пусть для определенности в миллиметрах.
4. Корабли за один шаг (одну команду диспетчера) могут перемещаться на ограниченное расстояние (не больше 50 мм) и поворачиваться на ограниченный угол (не больше 5^o) относительно центра локальной системы координат модели.
5. Диспетчер на каждом шаге выдает команды всем кораблям одновременно. Если корабль уже приземлился, и должен оставаться на месте, то в последующих командах для него нужно указывать одно и то же финальное положение.
6. На каждом шаге корабль поворачивается вокруг заданной оси на заданный угол и перемещается вдоль заданного направления равномерно (https://en.wikipedia.org/wiki/Slerp)
7. В процессе посадки корабли не должны столкнуться друг с другом:
    a. *расстоянием между кораблями* назовем минимальное расстояние, на которое нужно сдвинуть корабли, чтобы их геометрии пересеклись;
    b. корабли не должны сближаться ближе, чем на 10^{-4} мм.

8. После приземления корабли должны быть вертикальны (в локальной системе координат ракеты вертикальная ось ориентирована по оси Z) и выстроены в одну линию (проекции центров масс на плоскость космодрома должны лежать на одной прямой параллельной оси X глобальной системы координат на равном расстоянии друг от друга; расстояние между соседними дюзами не должно превышать 10 мм) и располагаться по росту (в порядке неубывания линейных размеров). Ось X локальной системы координат должна быть параллельна либо оси X, либо Y глобальной системы координат. Все направления сравниваются с точностью до 0.1°



9. После посадки корабли должны располагаться на плоскости космодрома ($z = 0$ в глобальной системе координат). Хотя бы одна точка ракеты должна находиться на плоскости космодрома (с точностью 10^{-4} мм).
10. В процессе посадки корабли не должны сталкиваться с плоскостью космодрома.
11. Число команд диспетчера не должно превышать 10000.



Входные данные

Набор 3D-шейпов кораблей в формате OBJ (https://en.wikipedia.org/wiki/Wavefront_.obj_file) и *трансформов* для них.

Входные файлы

- h-rocket-1.obj ... h-rocket-N.obj - файлы с 3D-шейпами кораблей;
- space.txt - файл с набором трансформов описывающих начальное расположение кораблей.

Формат OBJ

Для упрощения формат используется со следующими ограничениями:

- описываются только координаты вершин и индексы полигонов (не используется информация о нормалях, текстурах, кривых);
- в качестве полигонов используются треугольники.

Simplified OBJ format sample

```
v 0.0 0.0 0.0
v 100.0 0.0 0.0
v 0.0 100.0 0.0
v 0.0 0.0 100.0
g group
f 1 2 3
f 1 2 4
f 1 3 4
f 2 3 4
```

Трансформ

Описывается информацией о повороте и перемещении:

[угол (w) в радианах] [ось вращения (a b c)] [вектор перемещения (x y z)]

Ось вращения описывается единичным вектором.

w	a	b	c	x	y	z
0.1	1	0	0	10.0	20.0	30.0

Трансформ описывает оператор A_n перехода из локальной системы координат объекта в глобальную на шаге $\boldsymbol{n}$.

Формат входного файла space.txt

- В первой строке задано число кораблей N, участвующих в посадке.
- Последующие N строк описывают файл с геометрией и начальное положение корабля *[имя файла с шейпом корабля] [трансформ (w a b c x y z)]*

```
4
h-rocket-1.obj 2.70094 2.84877e-005 0.0111434 0.999938 2.18222 -1596.8 2495.6
h-rocket-2.obj 1.23534 0.0500927 0.029307 0.998314 -224.393 -2287.96 3316.76
h-rocket-3.obj 1.99174 0.156216 0.249431 0.955709 444.722 -1827.9 3779.7
h-rocket-4.obj 3.02753 0.00335115 0.00113511 -0.999994 681.413 -2276.26 2870.48
```

Гарантируются, что входные данные подобраны так, что в начальный момент корабли не сталкиваются, находятся в полупространстве $z > 0$. Компоненты вектора перемещения начальных трансформов по модулю не превосходят 10^4.

Выходные данные

- result.txt - файл с набором последовательных команд диспетчера (каждая строка описывает положение корабля после выполнения очередной команды)
- в рамках одной команды диспетчера должна соблюдаться последовательность выдачи задач кораблям – от *1* до *N-го*

Формат выходного файла result.txt

номер корабля [трансформ (w a b c x y z)] после выполнения команды

```
1 2.70094 2.84877e-005 0.0111434 0.999938 -3.79555 -1586.38 2538.08
2 1.23534 0.0500927 0.029307 0.998314 -236.203 -2295.62 3359.24
3 1.99174 0.156216 0.249431 0.95571 450.775 -1823.32 3822.18
4 3.02753 0.00335115 0.00113511 -0.999994 693.147 -2283.6 2912.96
...
```

Разница трансформов двух последовательных положений корабля $A_n^{-1} * A_{n+1}$ должна удовлетворять ограничениям, описанным выше. Число строк в выходном файле кратно числу кораблей N.

Оценка результатов

- За 1 час до окончания соревнования выдаётся до 100 дополнительных вариантов входных данных space-002.txt, space-003.txt и т.д. (000 и 001 известны заранее).
- Для каждого варианта участники вычисляют последовательность команд result-000.txt, result-001.txt и т.д.
- В первую очередь участники оцениваются по количеству решенных вариантов, удовлетворяющих всем ограничениям. При равенстве сравнивается среднее количество команд диспетчера, затраченное на посадку.

Визуализация траекторий

Для визуализации траекторий движения ракет, предоставляем python-скрипт "compute_trajectory.py", принимающий на вход файлы "space.txt" и "result.txt". На выходе скрипт создает набор файлов, где каждый файл содержит кадр движения моделей заданного в "result.txt". Формат файлов - VTK (https://www.vtk.org/). Они могут быть открыты в любом приложении для 3D визуализации, поддерживающем данный формат, например, Paraview (https://www.paraview.org/download/).

Paraview имеет возможность открытия и последовательного просмотра группы кадров, нахождения сечений объектов (может быть полезно, например, для изучения регионов столкновения) и пр.