

Создаем объект Ровер(класс)

Параметры ровера: -длина
-ширина
-высота
-диаметр(радиус) колеса

Все параметры ровера задаются в пикселях.

Далее используя данные параметры находим два угла наклона ровера (вертикальный и горизонтальный), при которых наш ровер будет заваливаться. Соответственно нам понадобится метод (или отдельно вынесенная функция), который(ая) в (дискретный) момент времени t будет рассчитывать угол наклона ровера и проверять завалился он или нет.

В каждый момент времени t ровер будет двигаться со скоростью 1 или со скоростью 2 (то есть двигаться за дискретный момент времени t на один или два пикселя соответственно)

Ровер будет двигаться в разных направлениях, задаваемых вектором скорости. Если мы хотим сделать крутой разворот, то мы останавливаем ровер, а затем продолжаем движение в нужном направлении. Для движения на один пиксель будет иметься 6 направлений, для движения на два пикселя 12 направлений.

Возникает необходимость в реализации новых команд, которые будут храниться в командном файле и которые будут прописываться в конфигурационном:

1)Остановка

*Если и только если скорость ноль мы можем развернуть ровер на 90 или 180 градусов

2)Вектор скорости (предположительно два параметра скорость и направление)

*Каждая команда переводит ровер в новый момент времени с новой точкой фиксации (фиксируем и считаем углы наклона, чтобы проверить жив ли наш ровер при таком муве)

3)Команда указания точек начала и конца пути, по которому будет двигаться наш ровер

*Наша задача будет состоять в том, чтобы построить путь (при заданных точках начала и конца), при котором ровер будет двигаться с максимальной скоростью

*При движении ровера мы запоминаем его путь, чтобы затем отразить его на общей картинке в gnuplot.

Отчет о семинаре за 07.10

Задачу с boundary и contrl пока не решил. С последнего занятия добавил бревна.

Программа скомпилировалась не сразу, ошибка была в неправильной реализации конструктора класса (поля класса и они же в конструкторе были заданы в разном порядке). После исправления программа заработала исправно.

Данная сказка уже находится на моем github