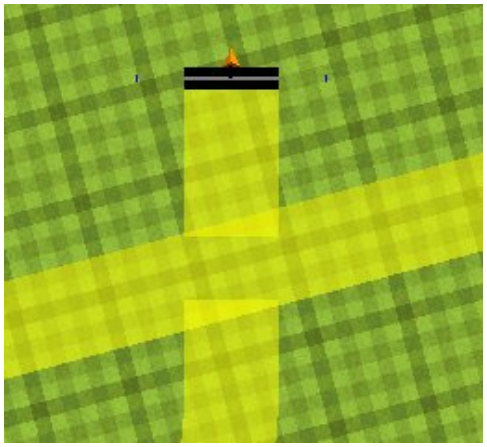
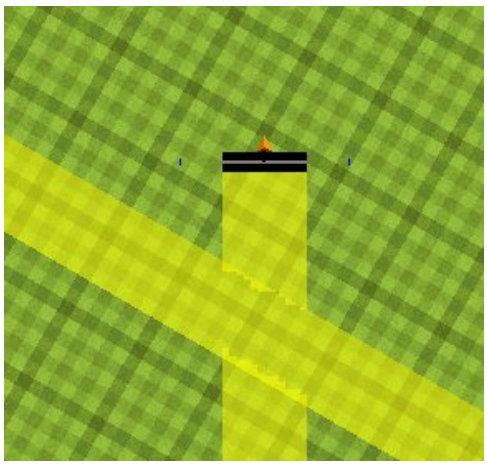
Scenario

You've just joined Trimble and are working in a team on a large monolithic application. The software controls how weed killer is applied to a field for a Tractor driving around with a spray implement attached. The spray implement has 12 nozzles across it's span, that can be individually turned on or off by the application. Each nozzle sprays weed-killer when on, and nothing when off. The application tries to ensure that weed killer is only applied once to any area of the field.

Your first task is to control the individual spray nozzles by turning them on and off based on the Tractor position and knowledge of where spray has already been applied. Currently the spray implement just turns all the nozzles on or off at the same time. You have to update the CImplement class to add this extra behaviour.



Current Behaviour: All of the 12 sections turn on and off at the same time (like there was only one spray section). As such, there is excessive overlap area when the tractor crosses an area where it has already sprayed.



Desired Behaviour: All of the 12 sections turn on and off individually. This provides fine grain control over where the weed killer is sprayed, and reduces overlap.

Task

- Update the source files to implement the new feature mentioned above
- Add new class(es), and update existing ones as you see fit to achieve the objective
- Comment where necessary, and correct any flaws you observe on the way (incorrect comments, functional issues, etc.)
- Use the API provided in the source code as much as possible, and enhance if necessary
- It is perfectly acceptable to add 'stub' methods that perform some useful function to help implement the solution, rather than implement a complicated well-defined algorithm. In these cases the purpose of the stub should be explained, and a description of how it would achieve that goal mentioned.