

# DART Field Editing

---

Pau Cano Ribé



## Document Changes

1.1	28/10/2020	Added Table of Contents, figure numeration, Document Changes Table. Small format changes
-----	------------	--

**Table of Contents**

1. Introduction..... 1

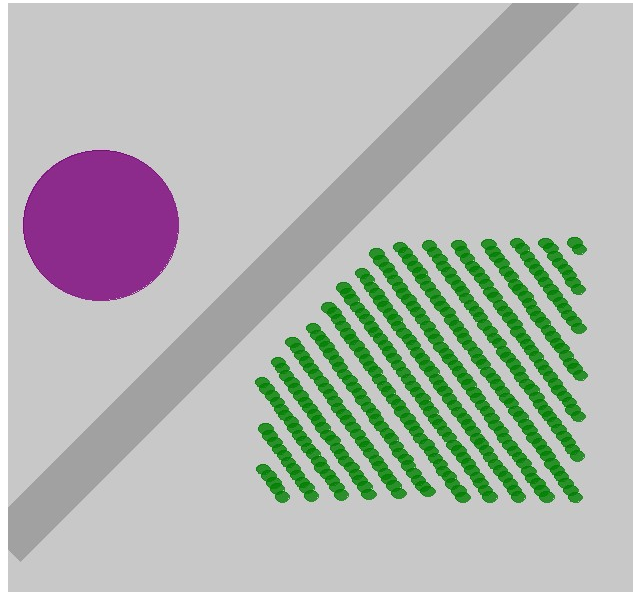
2. Automatic clearance of fields..... 2

3. Requirements ..... 5

## 1. Introduction

DART's built-in tool for field creation (found in Tools → Python Script Launcher → Select → CropFieldCreation → multi\_field\_generator.py) is versatile, allowing to create fields with different shapes, objects, and settings easily and quickly. But, it does not allow for cleared sections of fields, which could be used to add singular trees, small water retention devices, or ponds.

Take into account the next simulation, with a vine field, a road, and an olive tree:



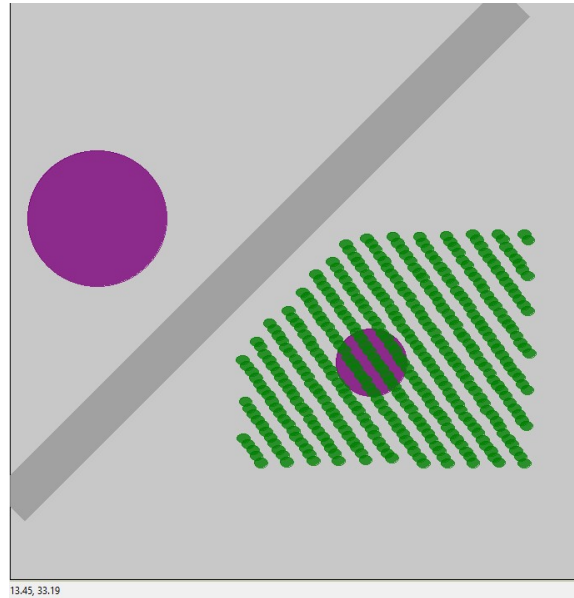
**Fig. 1.1:** Simulation overview in the parameter window, which shows an olive tree in purple, a field in green, and a road in dark gray.

Imagine we wanted to add another tree in the middle of the field, something like in this image:



**Fig. 1.2:** Aerial image of a field with three trees between some rows.

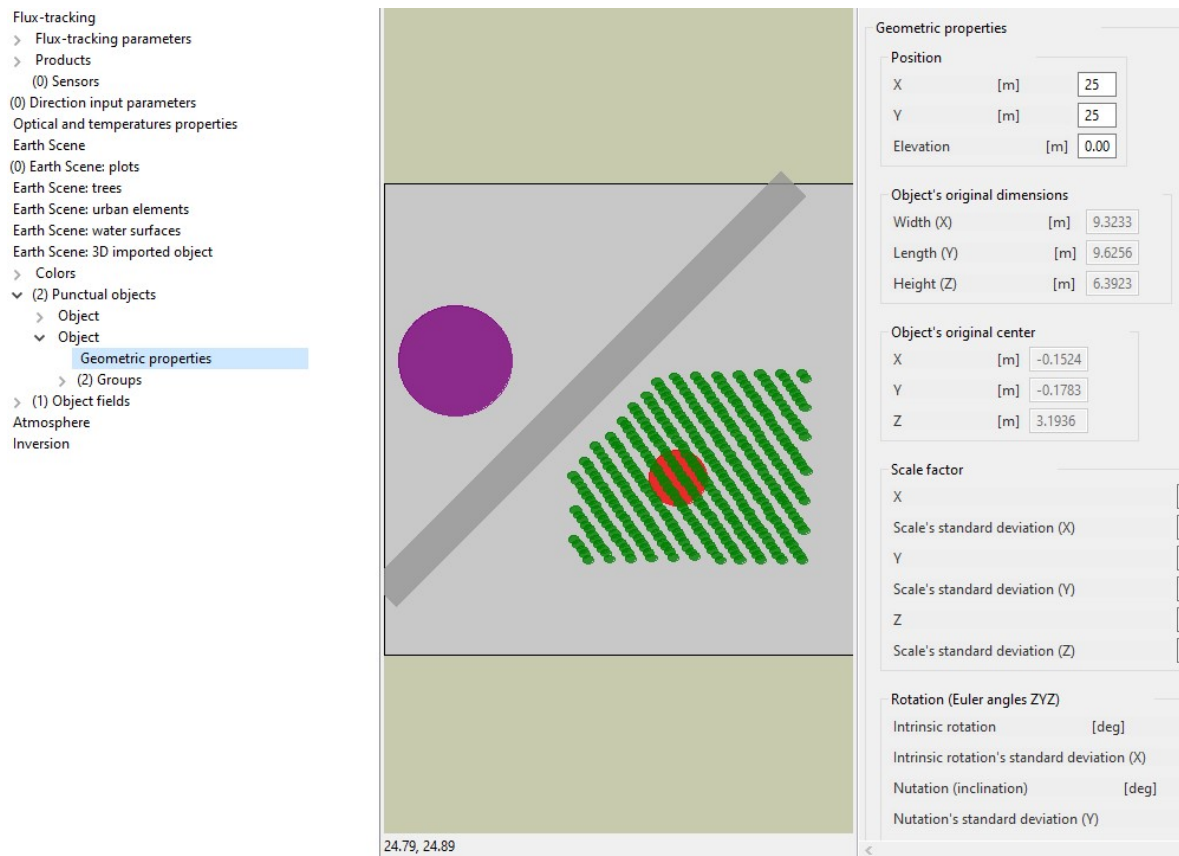
If we try to do this directly in DART, we end up with the tree right above the vines, and no way to clear that specific section of the field, unless we delete manually each of the object descriptions in the field file:



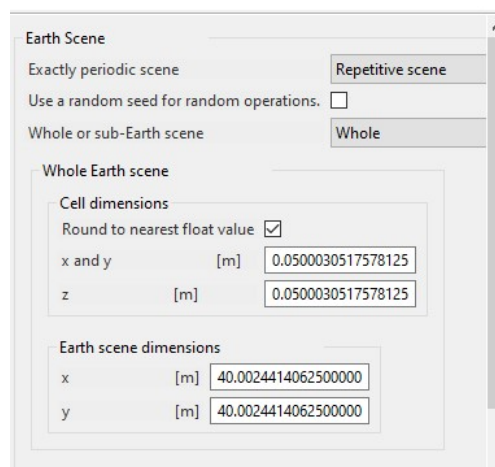
**Fig. 1.3:** Simulation overview showing the added olive tree, in purple, inside the field.

## 2. Automatic clearance of fields

In cases like this, if we know the size of the scene (set up under Earth Scene, as seen in figure 2.1), and the coordinates and size of the object, which we can take from the object properties as seen in figure 2.2, or by looking at the coordinates in the bottom part of the viewing window, we can create a map of the places we can to clear.



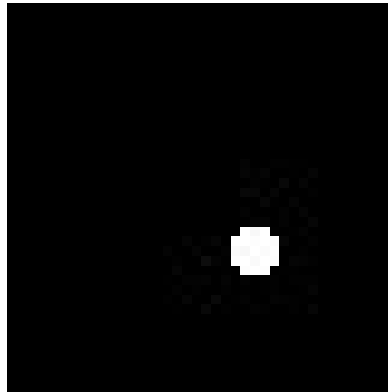
**Fig. 2.2:** Simulation overview showing the selected olive tree, in red, inside the field. The object location can be seen in the top part of the right window, under *Geometric properties*. The position of the object can also be seen in the coordinates that appear in the bottom left part of the scene preview, which show the position of the cursor. When the mouse is hovered over the object, the coordinates are, approximately, that of the object. In this case, (24.79, 24.89) instead of (25, 25).



**Fig. 2.1:** Earth Scene parameters. The dimension of the scene can be found under *Earth scene dimensions*. In this case, 40m by 40m.

In this case, we want to clear a 5mx5m round area centered at (25, 25). As such, we can draw a map (can be done easily in Paint, using a black background and the pencil tool with color

white), where each pixel represents  $1\text{m}^2$ , and white represents areas to clear. The size of the image should be the same as the size of the scene. In this case, 40x40:



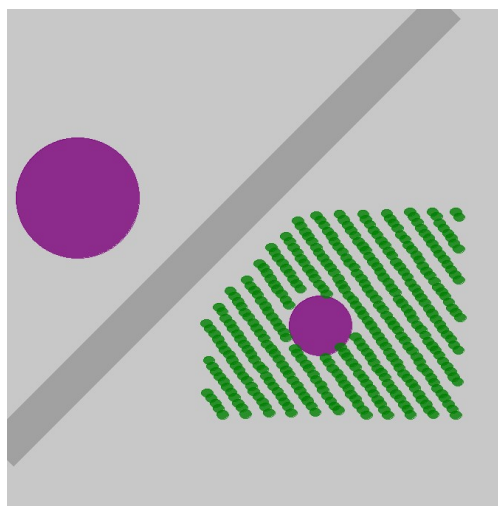
**Fig. 2.3:** Black and white image, with areas that should be cleared from the field in white. Each pixel represents  $1\text{m}^2$ .

Once we have this map, we can clear automatically these areas of the field by calling the file `clearSections.py`. The general call is:

```
python clear_sections.py -f field.txt -i image.png
```

The parameters passed to the script correspond to the field text file used to describe the field, which will be located where you saved it after creating it in the field editor, and the image map where the areas to clear are painted. For more information about the field editing tool, refer to section 3, *Field Creation*, of the Basic DART Guide.

After executing this script, with the correct parameters, you should reopen DART, or reassign the field file to view the changes. The old field file is also saved, with the name modified to `field_YYYYMMDD_HHMM.txt`, where the letters correspond to year, month, day, hour, and minutes, respectively.



**Fig. 2.2:** Simulation overview showing the added olive tree, in purple, and the area cleared in the field.

This process can be repeated as many times as wanted.

### **3. Requirements**

Python must be installed, and the skimage module able to be used.