Data Structures and Algorithms Assignment 1

Daniel Knopp

The daily life activity I chose to explain via a flowchart is mowing the lawn. Typically, I must complete this task once a week and I use a push mower (although I tried to write the flowchart in a generalized way not specific to the style of mower being used). I did, however, use language specific to an electric mower – but this language could easily be converted to gas powered mowers by checking fuel levels instead of battery levels and filling the fuel tank instead of charging batteries.

I made sure to explain each aspect of the procedure in explicit detail, but some of the axillary tasks (such as what to specifically do with the infinitely many possible objects one might find in the yard before mowing) I generalized to keep the flow chart from getting unnecessarily massive. All parts directly related to the act of mowing the lawn though were outlined in great detail.

Start

Divide the yard into regions ('Front', 'Back', etc.) and input region list

Check Mower Battery Levels

Do batteries need to be charged (<50%)? — Yes → Insert batteries into charger → Wait 10 minutes

Diagram Key
- Yes
- No

Are batteries charged (==100%)? — No → (back to Wait 10 minutes)

Are batteries charged (==100%)? — Yes → Set yard_region to 1 and head to yard_region

Do batteries need to be charged (<50%)? — No → Set yard_region to 1 and head to yard_region

Scan yard_region for any large sticks or other yard debris

Does any debris exist? — Yes → Count num_debris_objects and set debris_object to 1 → Retrieve debris_object → Is debris_object garbage/yard waste? — Yes → Is debris_object too large for trash can? — Yes → Break and/or dismantle debris_object into pieces small enough to fit

Is debris_object garbage/yard waste? — No → Put away debris_object

Is debris_object too large for trash can? — No → Place debris_object and/or all pieces in trash can

Break and/or dismantle debris_object into pieces small enough to fit → Place debris_object and/or all pieces in trash can

Put away debris_object → Increment debris_object
Place debris_object and/or all pieces in trash can → Increment debris_object

Increment debris_object → Is debris_object > num_debris_objects?

Is debris_object > num_debris_objects? — No → Retrieve debris_object

Is debris_object > num_debris_objects? — Yes → Generate a path for the remaining unmowed_perimeter of yard_region

Does any debris exist? — No → Generate a path for the remaining unmowed_perimeter of yard_region

Are there any fixed_objects in the way? — Yes → Count num_fixed_objects and set fixed_object to 1 → Adjust remaining path inward to go around fixed_object in a smooth spline curve → Execute path until deviation around fixed_object from the original path is complete → Is there any unmowed lawn remaining around fixed_object? — Yes → Mow unmowed region around fixed_object

Is there any unmowed lawn remaining around fixed_object? — No → Increment fixed_obect

Mow unmowed region around fixed_object → Increment fixed_obect

Increment fixed_obect → Is fixed_object > num_fixed_objects?

Is fixed_object > num_fixed_objects? — No → Adjust remaining path inward to go around fixed_object in a smooth spline curve

Is fixed_object > num_fixed_objects? — Yes → Execute remaining portion of unmowed_perimeter

Are there any fixed_objects in the way? — No → Execute remaining portion of unmowed_perimeter

Execute remaining portion of unmowed_perimeter → Has the yard_region been fully mowed?

Has the yard_region been fully mowed? — No → Shift mower inward by width of cutting path → (back to Generate a path for the remaining unmowed_perimeter of yard_region)

Has the yard_region been fully mowed? — Yes → Is yard_region == num_yard_regions?

Is yard_region == num_yard_regions? — No → Increment yard_region → (back to Check Mower Battery Levels)

Is yard_region == num_yard_regions? — Yes → Return mower to garage

End