CS 481 — AI — Capstone Project — Farmzoids

Capstone Project — Farmzoids
Introduction

The A.I. Capstone Project includes several A.I. techniques. This project is to develop a group of Farmzoids to perform planting, growing, and harvesting on a farm. The farm will be created on the large grid (40 by 40) you're already familiar with, but without a maze. A couple obstacles (river in upper left area (two bridges & you can also go around it at the grid edges.) You will provide the Farmzoids (4 bots) with the location of a Barn, which has the resources, equipment, and harvest storage needed. You will also provide the Farmzoids with the locations of the planting "plots" (you choose locations manually) into which plant seeds (3 kinds of seeds: Apple, Berry, Corn) will be placed.

The Farmzoids will attempt to grow crops while countering the actions of Nature. The objective is a large harvest at the end of a growing season. (Have room for 2-3 growing seasons.) (No penalty for stuff that dies.)

Both Nature (WMEM plants & environment) and the Farmzoids (WMEM has tasks to be done & maybe current zoid task) will be implemented using rule-based systems, and they will share a working memory (WMEM). Additionally, the Farmzoids will be equiped with a **Decision Tree** (build by hand; no info-gain stuff needed; no training sets are necessary) which will tell, from a plant's description (in WMEM), what it might need (water, soap, fertilizer, no-water?). (FMS may need to remove old un-run tasks at the start of each day.)

This project will be built using the same HTML-JS-P5-Lisp coding stack used for the warm-up. In addition to getting to demonstrable features, some attention needs to be paid to cleaned up and commented Lisp code (use pseudo-code (English-ish) sketch as code commentary– comments usually tell a) why you do it, or b) give you the bigger picture as to what is being done).

Nature

Nature will update the details of each plant: its look(seed, stalk, bush, blossom, green-fruit, and red-fruit), color (green, yellow, brown, black=died), water reserves (0 to 3 units, above is overwatered & the plant has a different color), age (till blossoming etc.), and current fruiting state (null?, blossom, green-fruit, and red-fruit/Harvestable, dropped). Nature will also engage some adverse effects: sky(clouds, rain, clear), wind (direction 4-winds), temperature (reflects clouds, rain, and wind),

and blight (one kind of blight per kind of seed) (courtesy of Nature, can affect neighbor bushes if the same kind of plant, use soap to cure). It will also maintain a day counter.

A day of clouds (:sky cloudy) delays plant growth that day. Clouds happen with a 10% chance each day. Rain (:sky rainy) happens with a 50% probability on each cloudy day. Each day of rain adds 1 unit of water to each plant's reserve. (Nature runs at the beginning of the day, before the farmzoids wake up; this can help avoid overwatering.) (Avoid coding up the Nature weirdness till normal stuff is up and running.)

A blight on a given plant can happen with a 2% chance each day. If it happens, the plant's growth is stalled until the plant is washed with soap (which is soapy water, pre-built) (which is available in the barn).

Wind comes from one of 4 directions: N, S, E, or W (each dir equally likely); like (:wind N). Wind happens with a 30% chance for each day, but doesn't affect things by itself.

A cold snap (:sky cloudy) or (:sky rainy) and a wind, like (:wind N), happens with a 50% change if the prior day was cloudy or rainy. A cold snap (Nature will preserve its environ state for the 2nd day.) (needs protection on second day – so we need zoids to notice and handle it – with smudge pots) lasts two days. If the plants **aren't protected** by the second day, then the unprotected plant crop has cold issues. To protect the plants, the farmzoids can place 3 smudge pots (there are 3 in the Barn) on the upwind side of the farm to protect the plants that are downwind within the 90 degree "cone" from the smudge pot's cell. (two 45 diagonals is a 90 deg cone. – may affect how you want to location the plots.) (Nature handles plant damage.) (Nature needs to monitor if smudge pots are placed & where.)

Plants

A seed must be planted in a "plot" which can be moved from the Barn to the desired grid location. (One plot per grid cell.) Plots must not have neighbors within two steps of one another (in any of the 8 directions).

A plant grows from a seed (with water and fertilizer) into a green-colored stalk, then into a tall bush (or tree, but we will use "bush" throughout). Once the bush has matured, it will go through four fruiting stages: flowering, green, red, and none/dropped. During the red stage, the plant can be harvested.

For each plant, a water reserve must be maintained from 1 through 3. Nature reduces the reserve by 1 each non-rain day. If a plant has no water at the start of a day, then Nature "decrements" its color; the colors go from green to yellow to

brown to black/dead. If the plant is given water below the green stage, then it uses up 2 units to get back to yellow, and 2 units to get back to green. (You can water a distressed plant & use Dtree again to see if it still needs water.) Once a plant goes black, it drops all its fruit (of any kind). (You can replant the plot, if you want.) If the plant goes above 3 in water, then its color changes to straw, and fruiting is stalled till it comes back down to 3 (when it turns green again). (Plots & seeds can be put down at any time.)

Plant Cycle

From the day (call it 0) when an apple seed is planted (in a placed plot, one plot per grid cell) and watered and fertilized, it goes to a stalk in 1 day. Then, except for Natural delays, it takes 3 days to turn to bush, then 6 days to blossom, then 3 days to show green fruit buds, then 2 days to turn to red apples. To summarize the stages are (seed stalk bush flower green-fruit red-fruit). Once gone red, it will stay for 3 days before dropping. For an apple seed: (1 3 6 3 2) = 15 days. It is the same for a berry seed, except this: (1 3 4 2 1) = 11 days. Corn is similar but shorter (1 2 2 2 1) = 8 days. A seed has no color and won't die if not watered, but won't grow until it is watered.

Harvesting

A bush/tree produces a half-barrel of apples, berries, or corn. You can't mix berries with others in a barrel without damaging them. The growing season is 40 days long; after which, no more harvesting can happen. (I don't think I talked about how fast blight spreads – pbb to a neighbor same-kind each day.)

Barn

The barn holds the storage for the harvest and all the resources: water, the seeds, the fertilizer, the smudge pots, soap, and harvest barrels. You can also have a farmzoid load up water from the river. (What river? See below.)

Farmzoids

Farmzoids move in any of 8 directions one grid cell at a time. A farmzoid "knows" how to move from one location to another while avoiding collisions with obstacles: don't step on a plot, don't collide with another farmzoid, etc. . A farmzoid can make 50 moves per day (hopefully that is a lot, with 4 zoids). You can have 4 farmzoids running at a time.

A Farmzoid can carry an unlimited supply of one kind of farming resource unless otherwise indicated. A farmzoid can

carry ten barrels of fruit from plants at a time. To change what the farmzoid carries, it must go back and step into to the barn. To start things off, the farmzoids must get and place plots (at locations you've chosen ahead of time, and built-in to the program). A farmzoid can carry 20 plots at a time. Then add a seed and fertilizer to that plot (maybe with different farmzoids), and then add water to start the seed growing. A given farmzoid can carry all the needed seeds, or all the needed fertilizer, or water for 20 plots. A farmzoid can't carry more than one of seed, fertilizer or water at the same time, but three farmzoids can move in tandem to more easily do these three tasks.

A farmzoid should provide updates to the F12 JS console on the significant events in its currently running task, (i.e., not every step, but the key points.): taking on a task, moving toward its first goal location (eg, Barn to collect <resource>, a plant cell to do <task>)). Alternatively, this running commentary from the farmzoids could be output in the webpage's display in a text area.

A farmzoid should pick a task that takes a shorter amount of time based on the zoid's current location at the time its prior task has completed. One way to do this is to determine the estimated travel time to complete the task. This is a goal, and need not be strictly enforced.

Once the basic aspects of the behavior of Nature and the Farmzoids is working reasonably well, some thought can be given to interrupting a farmzoid in its currently running task in order to drop what it is doing and start an emergency priority task. This is a bonus option for extra credit. Care should be taken to add the dropped task back into the WMEM's ready tasks when dropping a task.

The Farm Monitoring System (FMS)

Nature

The Barn includes a farm monitoring system (FMS) that can look at Nature' WMEM state and create tasks from it. FSM should run at the beginning of each day, after Nature does its update, and before the farmzoids continue their prior day's tasks.

FMS can learn what a plant needs by running a decision tree (DTree), to classify the plant's current needs (and adds zoid tasks as needed). The classification should then be used to run a rule to determine what tasks need to be done for that plant. (Ideally, Nature, then FSM, then zoids run.)

The Farm

The farm is a 40 x 40 (or so, if you like) grid of cells (0,0 to

39,39). A 1-cell wide (blue) river runs from cell (1,25) diagonally to cell (15,38) with two two-cell wide bridges dividing the river length into 3 approximately even segments. Hence, there are 4 ways across the river. Also, there is a caved-in mine area (a black rectangle) with corners (14,26) and (18,29). The river and the cave-in are obstacles to the farmzoids, so they should not cross these areas. You can place the barn and plots anywhere else on the farm.

Rule-Based System

Nature has a set of rules which deal with assertions in WMEM. Nature runs her rules at the beginning of the day, before the farmzoids run. Here's an example rule:

```
;; Nature's Rules fragment.
;; Adjust plant color level.
(:rule n2 :if (and (:plant xx :have-water yy)
                   (:sky ss)
                   (:eval (not ss rainy))
                   (:eval (= 0 yy)))
      :then
      (decr-color (:plant xx :color))) ;; green to yellow to brown to black.
```

Here the rule #n2 checks if there is a plant with no water and the sky is not rainy; and if so, then "decrements" that plant's color field. The two :eval antecedents need their arguments "evaluated" to determine their truth. The :plant antecedent matches any plant and retrieves its :have-water slot value. The :sky antecedent is just matched to an element of WMEM, etc.

Here are some more examples.. These fragments are notional only. If they seem to need adjustments, please feel free to do so.

```
;; Nature's WMEM fragment.
(:sky clear)
(:day 12)
(:wind north) ;; Wind from the North
(:temp warm) ;; or cold
(:plant (3 3) :kind berry :look bush :color yellow :fruit green :age 8)
(:plant (7 9) :kind berry  :look bush :color green :fruit red :age 5)
(:plant (8 10) :kind apple  :look bush :color white :age 2)
(:plant (6 10) :kind apple  :look stalk :color straw :age 1)
(:plot (5 2)) ;; Awaiting a seed via (:do seed).

;; Zoid Tasks WMEM fragment.
(:task 1 :do plot :loc (4 10))
(:task 48 :do soap :loc (8 10))
(:task 10 :do water :plant (3 3))
(:task 10 :do harvest :plant (3 3))
(:task 115 :do seed :loc (3 3) :kind berry)

;; Zoid Rules fragment.
(:rule 11 :if (and (:bot zz :at xx)
                   (:plant xx)
                   (:bot zz :task water)
                   (:bot zz :have-water yy))
       :then
       (incr-water (:plant xx)) ;; top off the plant's water.
       (decr-water (:bot zz)))

;;FMS Rules fragment.
(:rule 14 :if (and (:plant xx :color yy)
                   ;; :check run DTree on :plant at xx.
                   (:check xx (has-blight yy)))
       :then (add-task :do soap :plant xx))

(:rule 33 :if (and (:plant xx)
                   (:check (needs-water xx)))
       :then (add-task :do water :plant xx))

(:rule 41 :if (and (:plot xx))
       :then
       (add-task (:do seed :plot xx)) ;; cvt plot to plant.
       (add-task (:do fertilize :plot xx))
       (add-task (:do water :plot xx)))
```

Team

The team size is the same as before, but you can change team members from the previous project if you wish (but do it soon).

Technical Debt

We emphasize working S/W (Rule #0). However, getting to working S/W fast often leaves technical debt – ugly code that is both hard to understand and complicates future modifications. Technical debt will rapidly turn into a Bad Smell if left too long to fester. Therefore, as this is the last project delivery, **the technical debt must be paid**, approximately in full; which is to say that your team's source code should be reasonably clean and well-documented (including reasonable code comments). As mentioned in the first lecture, a handy way to get reasonably good comments is to write your functions/methods in natural (eg, English) language and then add this pseudo-code as comments to

your actual code.

Academic Rules

Correctly and properly attribute all third party material and references, lest points be taken off.

Project Reports, Submission & Readme, Grading

Same as before.