HOMEWORK CLASS 22

Many decisions can be modeled as a process of identifying alternatives and selecting the one that best meets some objective. But in reality decisions are very often tradeoffs: alternatives that provide more of one objective very often provide less of some other objective.

Businesses may make decisions to increase profit or market share, but doing so often increases the risk of losses. Animals may seek higher food intake but must consider that eating more can also increase the risk of being eaten. Political parties may act to increase their popularity but at the cost of compromising their core values. Modeling these kinds of tradeoff decisions is one of the most challenging, and innovative and exciting, tasks of agent based modeling.

Adaptive behaviors can often be modeled using a process of

(1) identifying alternatives
(2) eliminating infeasible alternatives
(3) evaluating the feasible alternatives by how well each meets an objective function
(4) selecting the alternative providing the highest value of the objective.

When alternatives are modeled as agents, NetLogo's built in agentsets can be used as a set of all alternatives to start the subsetting process. If a turtle is selecting a patch to move to, then the builtin agentset **`patches`** is automatically a set of all possible destinations. If the turtle can only move to an adjacent patch, then the primitive **`neighbors`** returns an agentset of such alternatives.

Write **`let`** statements that create a local variable containing the following agentsets. For the first six, find ways (e.g., test programs, statements typed into the Command Center or Agent Monitors) to show that your statements work correctly.

1. All the green turtles (NetLogo's standard green color)
2. All the turtles colored any shade of green, from darkest to lightest
3. All the patches with at least one turtle on them
4. All the patches with no turtles on them
5. All the patches with exactly two turtles on them
6. All the turtles in the lower right quadrant of the space
7. In turtle context, all the turtles within a radius of 3 patches
8. In turtle context, all the turtles within a radius of 3, except the calling turtle (the turtle
9. executing the statement)
10. In turtle context, all the turtles that have directed links from the calling turtle

11. In turtle context, all the turtles that have undirected links with the calling turtle
12. In turtle context, all the turtles on patches neighboring all the patches that contain a
13. turtle with a directed link from the calling turtle
14. In patch context, all the turtles within a radius of 5 patches of the patch’s center
15. In patch context, all the turtles on patches within a radius of 5 of the patch’s center