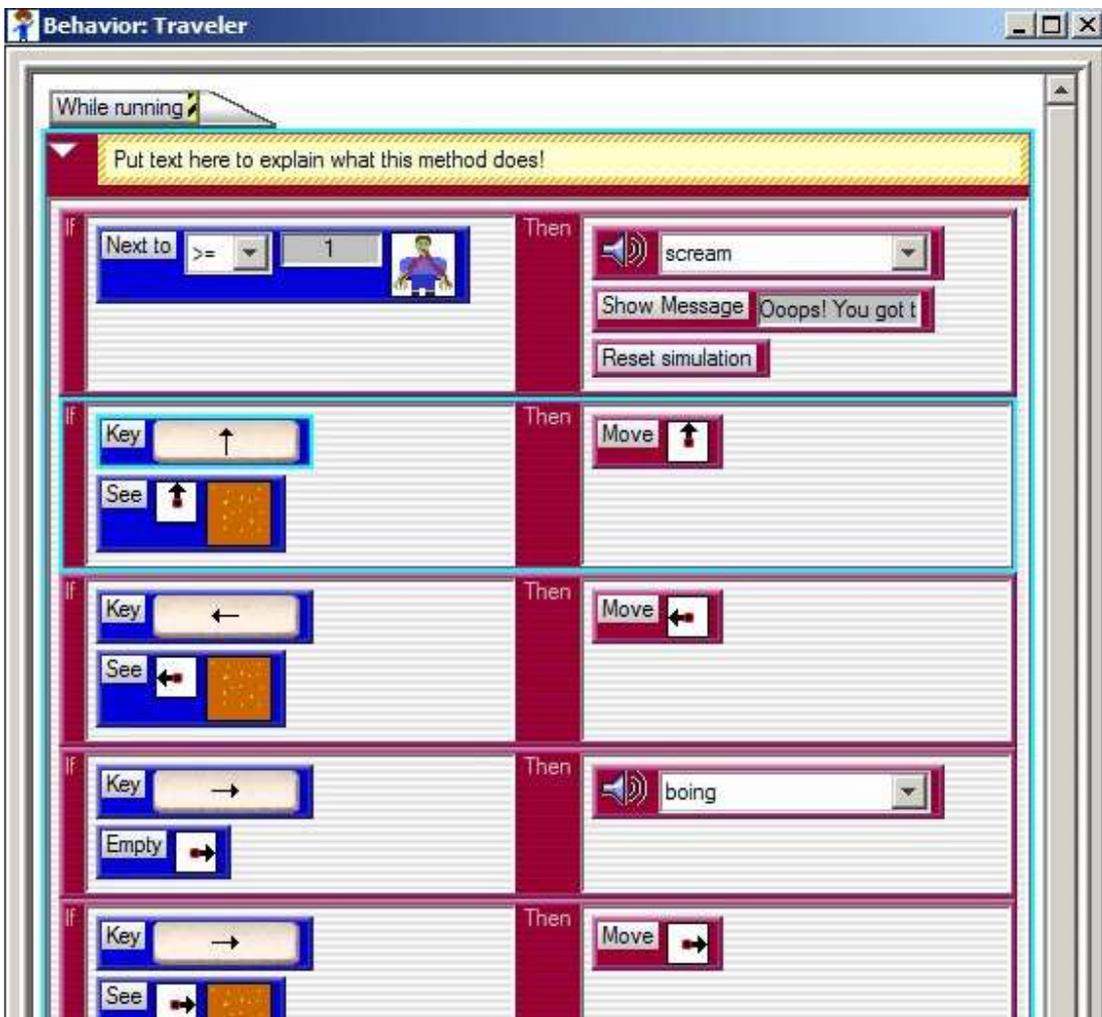


Journey Project Summary Part III – Polling, Simulation Properties

Overview: In this enhancement to the Journey project, the traveler must “collect” – that is get next to – multiple goals in order to win. In addition, we alter the way in which the end of the game is determined from a simple single event (getting next to a goal) to detecting when all of the goals are reached. To accomplish this, we introduce the concept of a simulation property, which is information that is shared among all agents in a project. We create a new agent, the “controller” to manage the process of polling the goals to determine when they are all “collected”; that is, when there are none left on the worksheet. Finally, we must change the behavior of the traveler agent so that it no longer declares the game is over when it gets next to a goal.

Traveler behavior change: This is a simple step: simply remove the rule that declares the game is won when the traveler gets next to a goal. So the new traveler behavior is now:



(Note: This is only a partial view of the traveler behavior, indicating that the game ending rule for winning has been removed.)

Controller agent behavior: There are two parts to the controller behavior. The first part is generating the information to make the decision regarding the end of the game. The second part is making the decision. Note that conditions basically “test” whether something is true; actions perform something as a consequence. It is not possible to test something as part of an action. In order to accomplish this – in this case, deciding whether the game is won, we use a separate method, which is a separate set of rules to do the testing/evaluation we wish.

In the “While Running” method, the control first sets the simulation property “@goals” to zero. Then it broadcasts a signal to all goals. We will see what the goals do in response later. This broadcast is called polling. Finally, the controller calls upon the “check_won” method to determine whether the game is won. This is true only if there are no goals remaining, which is determined by the @goals simulation property being zero. If any goals are left, we will see that this simulation property will be greater than zero.

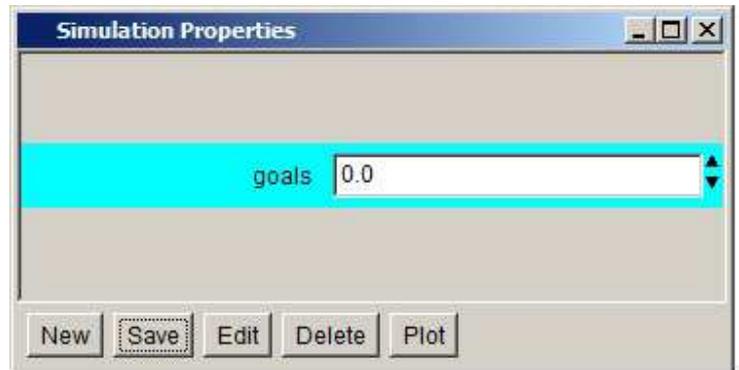


The screenshot shows a software window titled "Behavior: Controller" with two rule methods defined:

- While running:** The condition is "Once every 0.3 Secs". The actions are: "Set @goals to 0", "Broadcast Goal", "check_in", and "Make check_won".
- On check_won:** The condition is "@goals is = 0". The actions are: "hallelujah" (with a speaker icon), "Show Message @goals and you win!", and "Reset simulation".

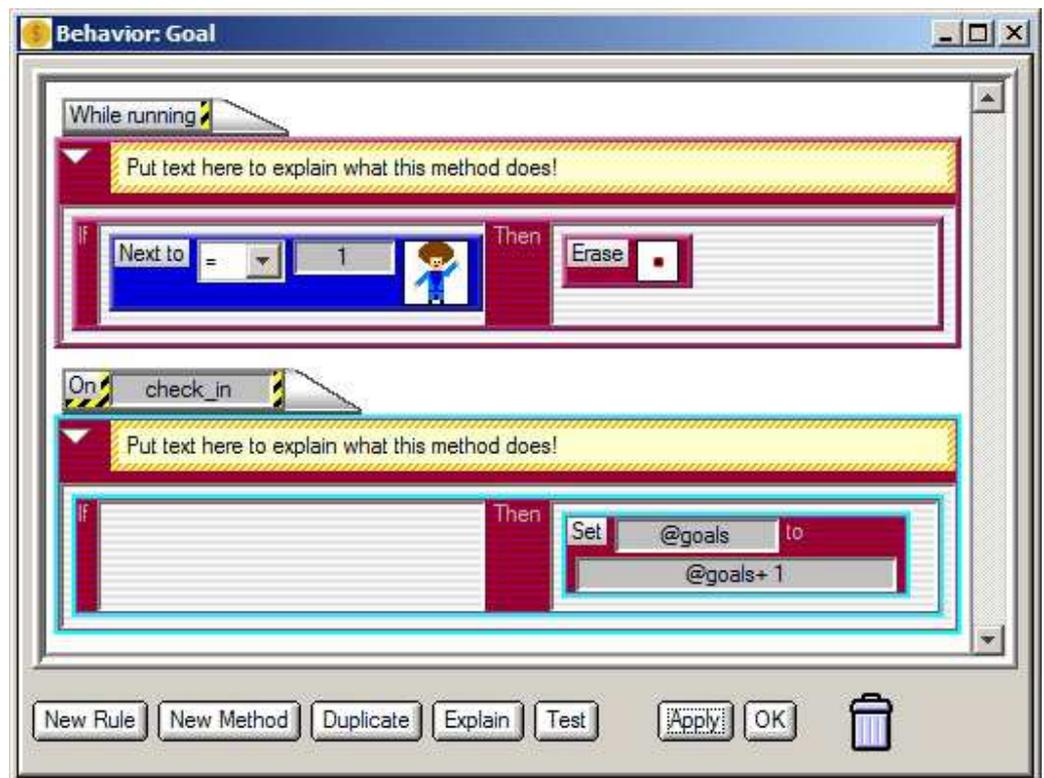
At the bottom of the window are buttons for "New Rule", "New Method", "Duplicate", "Explain", "Test", "Apply", "OK", and a trash icon.

Setting the initial Simulation Property: In many cases, such as this project, it is actually not necessary to create and set a simulation property in advance of using it. AgentSheets will automatically create a simulation property the first time it is referenced and set it to zero, which, in fact, will work for this project. However, it is good design and programming practice to “declare” such information as simulation properties as part of the project/program construction. To do this, pull down the Simulation Properties window from the Tools menu. The initial value of 0.0 is fine. Click ‘Save’ to preserve the new simulation property.



Goal behavior changes: There are two behavior changes required for the goal agent. The first is to handle being “collected” by the traveler agent. (Note: this is one way to handle this situation; there is a more elaborate way technique that could be placed in the traveler agent. In more advanced projects, such as a predator-prey model where a predator “eats” a prey, such an alternative technique would be needed.) The goal agent now detects being next to the traveler, instead of vice versa. We simulate “collecting” the goals by having the goal erase itself when it detects proximity to the traveler.

 The second behavior change for the goal agent is to respond to the “poll” (broadcast) from the controller, to update the @goals simulation property. This second change is in the form of a separate method; it is not part of the continually running “While Running” method, since it only runs when called by the controller agent. Each remaining goal agent will increment the @goals simulation property. If no goal agents remain, then the @goals property will be zero, which the controller agent will detect and declare the game won.



Revised worksheet: We have placed one controller on the worksheet. Not that the controller agent does not need to be in the active area of the worksheet, since it does not interact directly with any agents. In addition, we have placed several additional goals on the worksheet, so that the traveler must “collect” – that is, get next to -- each of them in order to win the game.

