Debugging tips and techniques for AgentSheets and AgentCubes projects

There is no one specific step-by-step approach to debugging in general, nor in addressing problems with AgentSheets/AgentCubes projects specifically. However, there are some general guidelines one can use in approaching problems that can well serve the debugging process.

- What are you trying to do? It is important for the user to be able to explain what the intended behavior is, rather than simply describe what is happening that is not desired.
- Read the appropriate agent behavior; try to understand what it says before running the simulation. Computers do what they’re instructed, which may differ from what is desired. Computers read code, not minds!
- Reset the worksheet before exploring. Start from “ground zero” when investigating a problem. Failing to do this is analogous to entering a theater half-way into the film: you’re likely to have missed some critical elements.
- Close all behavior windows (look for minimized windows); this will indicate where there are behavior changes that have not been made permanent. One cause of unexplained behavior is that the user made changes to it, but failed to apply the changes by clicking on “apply” or “OK” before running the simulation.
- If “mysterious” agents appear, check the worksheet by removing one layer at a time using the block erase tool to determine if there are hidden agents. The single most common error in constructing worksheets is to treat the worksheet like a paint canvas and the agent placement tools like a paintbrush. The worksheet is a layered structure and the placement tools will place agents on top of each other so that the agents underneath are typically not visible. Teach students to use the multi-agent (rectangle) tool for placing rows, columns, or blocks of agents, and the pencil tool for placing individual agents.
- Close the project and exit/restart AgentSheets, especially if any kind of error message occurred, such as a Java error message. Even though AgentSheets appears to have recovered from an error, it is possible, if not probable, that some internal structure is not in a good state. Exiting the application and restarting it will ensure this is not the case. CAVEAT: if AgentSheets appears to be “hung”, then before terminating it, find and copy the entire project folder to a separate location. Especially on a Windows system, if AgentSheets is in the process of closing its files and the AgentSheets process is forced to quit (using the Task Manager), the main project file (“.apj” file) may be damaged. Although it is possible to recover most, if not all, of the project when this happens, it is by no means trivial, and it is not a “standard” documented technique. There is an FAQ, however, that describes this process in detail; however, it is not for the faint of heart!
- Rule order can be critical. Look for rules with no conditions. These rules will be performed unconditionally, which means any rules below such a rule will never be evaluated for execution. In general, rules that deal with “unusual” situations, such as game-ending rules, should be placed highest in the behavior list. As in life, first things first.
- Condition order can be critical. Normally, since all conditions of a rule must be true, one might conclude that their order is not important; just as the order of selection of a sports team is not important; rather only the final team composition is important. But there are situations in which this is not the case in AgentSheets. The most illustrative example is one in which there is a timing condition and a probability condition. For example, if the conditions are
Once every 1 second, with a 50% chance: This rule will be performed on the average about every 2 seconds.

However, reversing the conditions (with a 50% chance, once every 1 second), the rule will be performed every second.

The difference is due to how AgentSheets evaluates rules in the “While Running” method: every agent is scanned sequentially and then the cycle is repeated as fast as the computer can perform – hundreds of times per second or more. Since the 50% chance condition will occur about every other cycle, this means that the 50% chance will be true about every other pass through the agents. This is much more frequent than every second, so when the 1-second timer occurs, there is certain to be a 50% chance success within a fraction of a second of that event.

- Use conversational programming! This is a unique tool among software development environments and a very powerful tool, as well. Use conversational programming to determine which conditions are satisfied and whether associated rules will be performed as expected.
- Examine agent attributes and simulation properties, as appropriate.
- Use the single step button. Watch behavior unfold incrementally, rather than trying to keep track of what happens in real time.
- Avoid “playing the game” to examine a specific event. Rather, move agents into appropriate positions.
- Remove agents not involved in the problem to permit examining other agents without interference. For example, if there are six agents pursuing a single agent, and the problem concerns whether hill-climbing is working properly, temporarily remove all but one of the tracking agents so that you can focus on one agent at a time.
- For diffusion, check the diffusion formula carefully. Usually, a mis-entered formula will cause an error message and the input rejected, but not always. Sometimes the formula will pass syntax checking but not be correct semantically.
- For hill-climbing, check the rules carefully. As with the diffusion formula, the code may be syntactically correct but not semantically.
- Check spelling of agent attributes and simulation properties. Note that these are created dynamically if not created deliberately. For example, if the user defines a simulation property named “total” (properly referenced as “@count”), but then mis-types the name as “@0tal” or omits the “@” symbol, the simulation will run and dynamically create the additional name, but the behavior may not work as desired.
- Switching worksheets: the proper way to select the destination worksheet is to allow AgentSheets to open the file menu, rather than directly typing in the name. Doing the latter often leads to typos, such as omitting the file extension (“.ws”).