Investigation and Questions for lab. 2
How the Restaurant Agents Work

(csci201 - Wilczynski)

The base class Agent.java

1. What happens when an agent constructor is called?
2. How does the agent thread get created and started? What gets executed?
3. What does the run() method do?
4. The professor talks about the agent scheduler. Where is it?
5. In the base Agent, the scheduler has the following signature:
   protected abstract boolean pickAndExecuteAnAction();
   What does all those keywords mean?
6. In the professor's design, his actions did not return anything, but here when an action is called, it returns True. What's going on and what is stateChange all about?
7. How does the semaphore get full?
8. The professor didn't call stateChanged() in his design. Why not?

The Cook Agent

1. Why in *msgHereIsAnOrder*() does the code put the information into the list *orders* instead of just storing the data into some globals like *currentWaiter, currentTable, currentChoice*?
2. Suppose I did use globals, will clobbering always make it fail?
3. The animation action DoCooking() has a call to timer.schedule(), which takes two parameters, a call-back object and a time to wait before calling the call-back. How is that first parameter formed?
4. Could I have done this more classically? [Please laugh at this pun.] Please try to write the code using a regular class.
5. Why is the parameter to DoCooking() typed as *final*?

The Waiter Agent

1. The waiter agent has a list of customers of type MyCustomer similar to the cook's orders list. In the MyCustomer class is a state variable for keeping track of the customer. One of the states is NO_ACTION. Can you explain what that is about?
2. In many of the waiter message reception calls, one of the parameters is the customer agent pointer and in the code there is a search to find the appropriate MyCustomer instance. How will that look in the Waiter design?
3. What if there is NO customer in the list? What does the "design" do?
4. There is a lot of code to support how the waiter moves around the animation. All that A* stuff is quite confusing. How come none of it appears in the design?

The Host Agent

1. How does the Host find his waiters? How does a waiter find its host?
2. How does the host pick a waiter to assign a new customer?
3. What are Waiter indices?
4. What is all that *synchronized* stuff in the code?

The Customer Agent

1. The customer agent looks different from the other agents. It's rules are different, why?
2. What is the proper state machine formalism?
3. How should a state machine look inside an agent? [Hint: Message are event setting, while the scheduler implements the transition mechanism.]
4. What kind of things would cause the fsm quality of a customer agent?
5. Suppose after the customer told the host that he was hungry. Right now the customer waits until a waiter seats him. Suppose the host asked the customer if he want to wait. Would that still fit the fsm framework? If so, how? [sketch a solution]