# ARTIFICIAL INTELLIGENCE
## Third Practical Assignment

## *ANITA* WORKS IN A FRUIT STORE AND SEES TO HER CUSTOMERS

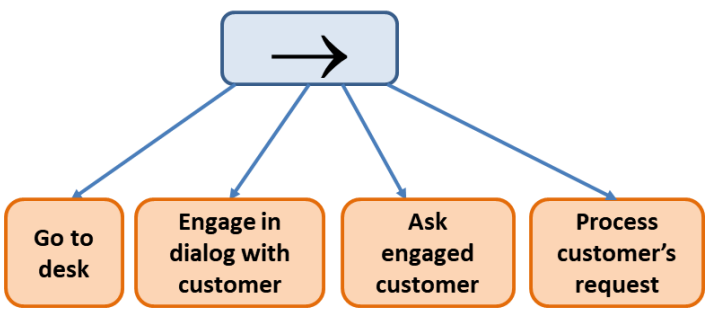
**PART A (MAX C+)**

**ANITA** is the main character in the scene that you have to complete. She works in a a very small fruit store, selling apples and peaches (and nothing more). While there's no customer, she sweeps the floor with her broom and sings marrilly. But when a customer pops in the store she sees to them. Her top-level behaviour can be easily modelled with a tree like the following:

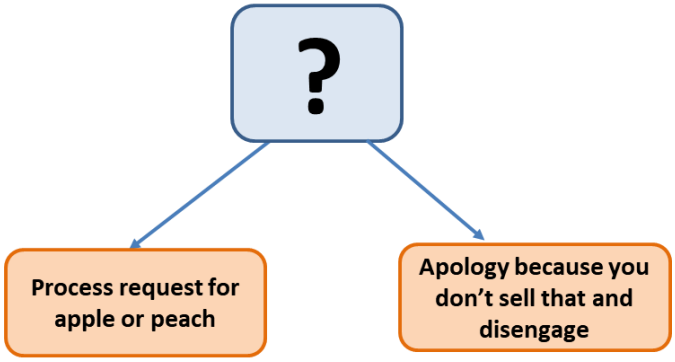

Sweep & sing is a simple behaviour. Anita just

- Clears her last utterance
- Produces her broom
- Produces her musical notes
- And then wanders around (an attractor) following a "constrained wander" approach that does not let her abandon the scene.

Seeing to a customer is something more complicated. It can be summarized by the following sequence:



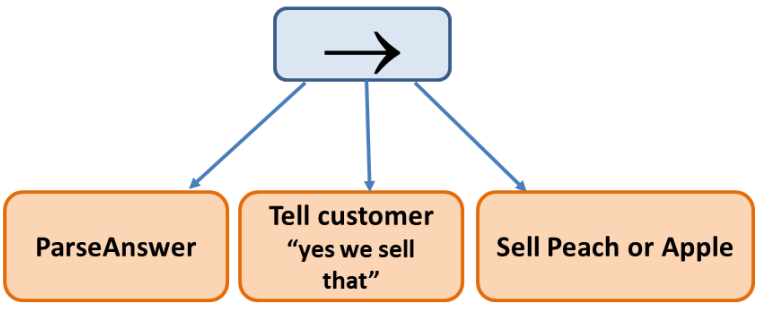
Enganging in a dialog and asking questions to an engaded customer (gameObject) are “primitive tasks”. Processing the customer’s request is the hardest part. Let’s see it more closely...




Left child processes (tries to process, actually) requests that have to do with peaches or apples (the only fruits Anita sells). If it succeds then the problem is solved. If it fails, Anita apologizes and disengages from customer (right child in the selector)

In order to apology, Anita uses the `TellEngaged` task. For disengaging she uses the `DisengageFromDialog` task.

Why should the left child [`Process request for apple or peach`] fail? Well, it fails because when the customer’s request is analyzed it does not contain neither the word PEACH nor the word APPLE. Hence, `Process request for apple or peach` is but a sequence that starts with the primitive task `ParseAnswer`[1]...

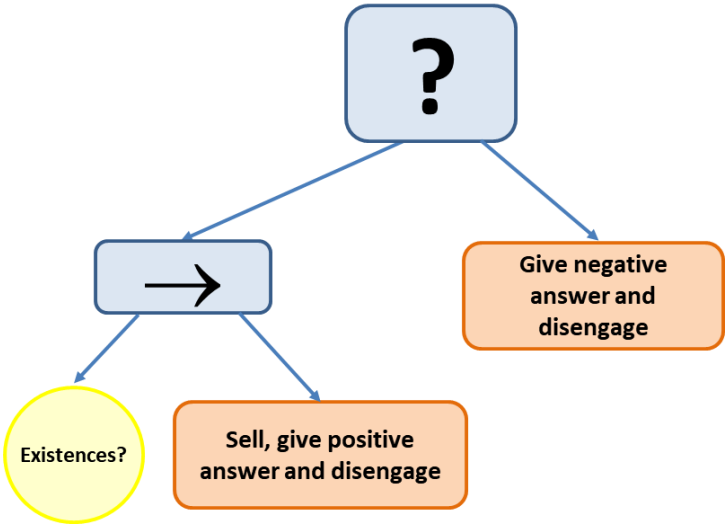



[Notice that “yes we sell that” does not mean that there’s any left. It just means that the requested product is one of the products sold in the store]

`Sell Peach or Apple` always succeeds (Yes, it even succeds if the required item is sold out).

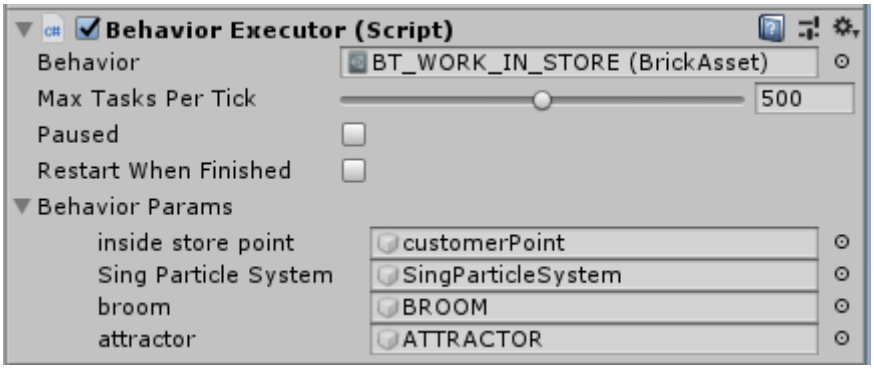
- If there are existences of the requested item: it is sold, the customer receives a positve answer and Anita disengages from them.
- If there are no existences left, the customer receives a negative answer and Anita disengages from them

[1] `ParseAnswer` is a task that fails if neither the word APPLE nor the word PEACH appears in the answer (input argument). If it succeeds the output argument contains the relevant word (APPLE OR PEACH)




**PRIMITIVE TASKS**
All primitive tasks have already been coded for you. For this part, you're unlikely to need new ones.
**BLACKBOARD**
Anita's blackboard implements a very simple dialog system.
**CUSTOMERS**
They're already coded. Their blackboards also implement a dialog system. That's how they interact with Anita. Have a look to their BT. You may find it instructive.
**USING TAGS**
Using tags in tasks like `ACTION_ArriveByTag` reduces the number of required arguments

Press C to and left-click to make a new customer appear



**PART B (GOING FOR A MARK HIGHER THAN C+)**

In order to get a mark higher than C+ you have to give ANITA a more complex behaviour (i.e. expand the behaviour developed in the previous part so that she can do "more things". This part is left to your imagination…

Some suggestions (they are just suggestions…)

- When the store runs out of merchandise (no apples and/or no peaches) she gets some more from a storage location (enlarge the scene if necessary).
- If instead of a customer she is visited by a thief (search suitable sprite) she cries "Police! Police!" until a policeman (or policewoman) shows up and arrests the thief.
- Make her utterances contextually correct but less rigid. For instance, to tell a customer that she has run out of a fruit, she may choose between:
    - Sorry, none left
    - Not a single one left, sorry
    - I sold the last one a minute ago
    - …

  (You may have to modify something in the customer so that they understand that's a negative answer)

In order to work on this part of the assignment the previous one must run without errors. Then you have to write a short report describing the new behaviours and showing the new BTs, if any (no screen-captures). If you have coded new tasks[2] (actions or conditions) list their names and briefly explain their purpose. Comment also on modifications affecting the blackboards.

[2] You are strongly recommended not to code new functions unless strictly necessary (for priority selectors, for instance). Instead use actions that fail or succeed.