

Spike Outcome Report

Number: 01

Spike Title: Goal Oriented Behaviour with Simple Goal Insistence

Personal: Peter Argent (7649991)

Goals:

Create a simple goal insistence (SGI) model simulation of goal-oriented behaviour (GOB) that demonstrates the both the effectiveness and the limitations of the technique.

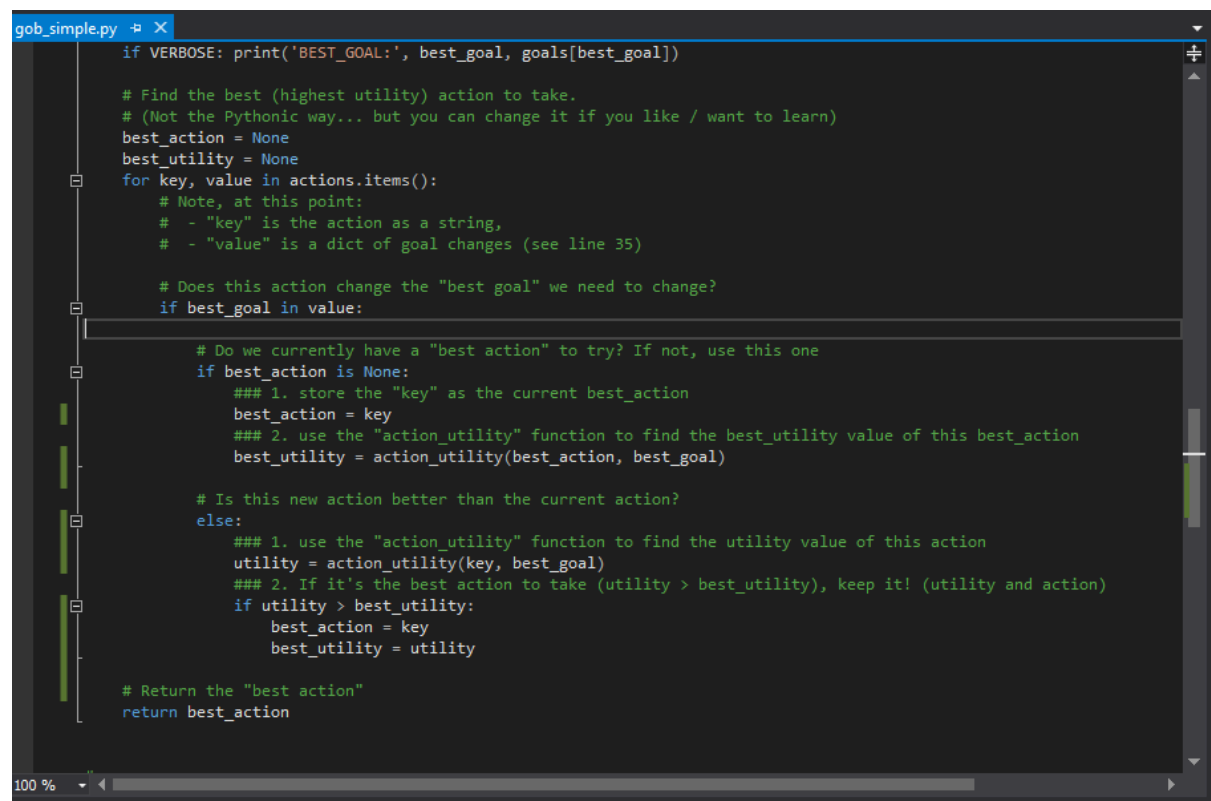
Technologies, Tools, and Resources used:

Visual Studio 2017 with Python 3 installed

Tasks undertaken:

Download Provided Code

Modify the choose_action() function to have get the best_action based of an action_utility() function that used simple goal insistence.



```
gob_simple.py
if VERBOSE: print('BEST_GOAL:', best_goal, goals[best_goal])

# Find the best (highest utility) action to take.
# (Not the Pythonic way... but you can change it if you like / want to learn)
best_action = None
best_utility = None
for key, value in actions.items():
    # Note, at this point:
    # - "key" is the action as a string,
    # - "value" is a dict of goal changes (see line 35)

    # Does this action change the "best goal" we need to change?
    if best_goal in value:
        # Do we currently have a "best action" to try? If not, use this one
        if best_action is None:
            ### 1. store the "key" as the current best_action
            best_action = key
            ### 2. use the "action_utility" function to find the best_utility value of this best_action
            best_utility = action_utility(best_action, best_goal)

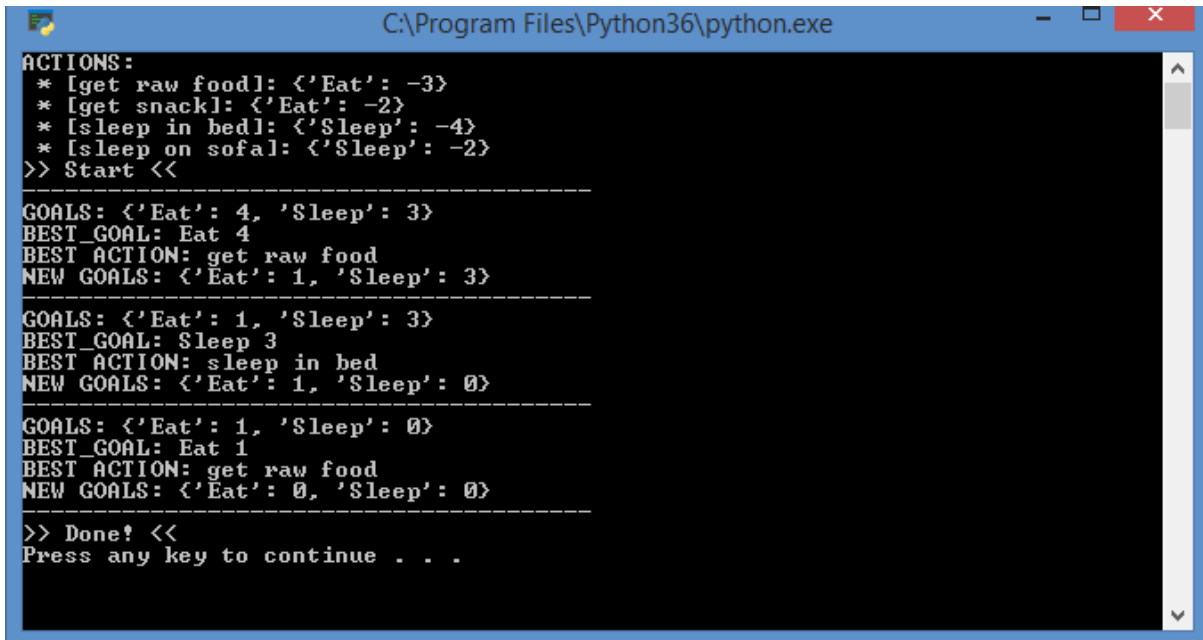
        # Is this new action better than the current action?
        else:
            ### 1. use the "action_utility" function to find the utility value of this action
            utility = action_utility(key, best_goal)
            ### 2. If it's the best action to take (utility > best_utility), keep it! (utility and action)
            if utility > best_utility:
                best_action = key
                best_utility = utility

# Return the "best action"
return best_action
```

SGI is a model that allows an agent to fulfil their need with the biggest insistence by using a utility function that calculates a value for each action that gets larger the better it fulfils the need.

What we found out:

SGI is great for simple AI that only wants its needs reduced. And will fulfil goals based on greatest insistence. It can be expanded by changing the action_utility function to care about other utility functions like a time utility or a resource utility. SGI becomes unwieldy the more goals and actions an agent has.



```
C:\Program Files\Python36\python.exe

ACTIONS:
* [get raw food]: {'Eat': -3}
* [get snack]: {'Eat': -2}
* [sleep in bed]: {'Sleep': -4}
* [sleep on sofa]: {'Sleep': -2}
>> Start <<
-----
GOALS: {'Eat': 4, 'Sleep': 3}
BEST_GOAL: Eat 4
BEST ACTION: get raw food
NEW GOALS: {'Eat': 1, 'Sleep': 3}
-----
GOALS: {'Eat': 1, 'Sleep': 3}
BEST_GOAL: Sleep 3
BEST ACTION: sleep in bed
NEW GOALS: {'Eat': 1, 'Sleep': 0}
-----
GOALS: {'Eat': 1, 'Sleep': 0}
BEST_GOAL: Eat 1
BEST ACTION: get raw food
NEW GOALS: {'Eat': 0, 'Sleep': 0}
-----
>> Done! <<
Press any key to continue . . .
```

Open issues/risks [Optional]:

Issue: The action_utility() function was given to us but is a simplistic utility that does not consider side-effects, time management or resource management in its calculation.