**Spike:** Spike\_1

**Title:** Spike\_GOB & SGI

**Author:** LE BAO DUY NGUYEN - 102449993

**Goals / deliverables:**

* Code see /spikes/spike04/
* Short report titled “Spike\_GOB & SGI”
* Completed gob\_simple.py

**Technologies, Tools, and Resources used:**

* Visual Studio Code
* Python 3.0+

**Tasks undertaken:**

* Download and install Visual Studio Code

A screenshot of a cell phone

Description automatically generated

* Download and install Python 3.0 & above

A screenshot of a computer screen

Description automatically generated

* Download and install Python extension within Visual Studio Code

1/ Go to Extensions

A screenshot of a cell phone

Description automatically generated

2/ Search “Python” then Install

A screenshot of a cell phone

Description automatically generated

* Add missing code into sample code

1/ Create a new file by selecting Start > New file

A screenshot of a cell phone

Description automatically generated

2/ Click on Plain Text then type in “python” to enable Python A screenshot of a computer

Description automatically generated

3/ Add missing codeA screenshot of text

Description automatically generated

4/ After having the code added, save the file by clicking this icon

A close up of text on a black background

Description automatically generated

* Compile sample code

1/ Right-click on the saved file, select “Open in Terminal”

A black and silver text on a screen

Description automatically generated

2/ Compile the code, by typing “python3 [saved file name].py”

A screenshot of a cell phone

Description automatically generated

**What we found out:**

def choose\_action():

'''Return the best action to respond to the current most insistent goal.

'''

assert len(goals) > 0, 'Need at least one goal'

assert len(actions) > 0, 'Need at least one action'

*# Find the most insistent goal - the 'Pythonic' way...*

best\_goal, best\_goal\_value = max(list(goals.items()), key=lambda item: item[1])

*# ...or the non-Pythonic way. (This code is identical to the line above.)*

*#best\_goal = None*

*#for key, value in goals.items():*

*# if best\_goal is None or value > goals[best\_goal]:*

*# best\_goal = key*

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

A screenshot of a cell phone

Description automatically generated

The effectiveness of SGI is that it allows the game bots to perform actions based on the insistence or goal varying on what it is. Referring to the output below, the best goal changes after an action, the higher the goal, the more it will influence the agent into making a certain action.

The disadvantage of using Simple Goal Instance is that it leaves little options for the bots to do anything afterwards (unless the goal was too simple). It doesn’t take into consideration the side effects because so far, there were only positive outcomes, to make it more convincing there needs to be negative effects as well, such as certain values to balance things out. Add some additional functions or values to make it more realistic.

**Risks/ Issues**

\_ Visual Studio Code issue when installing and running Python + its in-app extension.

\_ Python version needs to be updated to the latest and run using “python3 …” command line.