Task 4 – Spike: Goal Oriented Behaviours & SGI

Context:

Goal Oriented Behaviour (GOB) is a simple approach to creating game agents that can perform actions based on an overriding "goal" which gives their behaviour direction and purpose. However, it has limitations which need to be understood.

Knowledge/Skill Gap:

Developers need to be aware of the limits of simple goal insistence (SGI) when used to resolve action selection using goal-oriented behaviour (GOB).

Goals/Deliverables: [CODE] + [SPIKE REPORT]

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.

You will need to deliver (show your tutor) the following items:

- Working code that simulates and **displays GOB using SGI**.
  (There is code provided, but you don't need to use it if you don't want to.)
- You must demonstrate a situation where SGI **works** appropriately
- You must demonstrate a situation where SGI **does not work** well.

To show your outcome (spike gap "closure") you must create a:

- **Spike Outcome Report**. (Use template headings. Your report should include snippets of code and results of demonstrations.)
- Make sure you include your **git commit history**/messages in your spike outcome report!
- Save your report document in your repository (in the task folder) as well as uploading to Doubtfire the version you want checked by your tutor.

Note: The "Spike Outcome Report" is **always** required as a "deliverable" for a spike task. It will not be stated as a deliverable in later spike tasks as it is implicitly required.

Important: Make sure you commit to your repository as you go, and good commit messages so that when you include your git commit messages in your spike report it is meaningful and valuable. Support yourself with evidence!

Start-End Period: Week 3 – Week 4

Planning Notes:

- Use the basic code provided on BB (gob_simple.py), which uses simple methods and dictionaries for data, and add the missing action selection code. Look for helpful ### comments and fill in the blanks.

Extensions:

- Create an object-oriented version of the code model and present the advantages/disadvantages this provides to game developers (or designers).
- Create a console-based turn-based role-playing game (simulation) that demonstrates two NPCs in combat with each other and using this simple resolution method to selection actions, and the outcomes.