Spike: 8
Title: Goal-Oriented Action Planning

Author: Luke Valentino, 103024456

Goals / deliverables:

Create a GOAP that demonstrates the effectiveness of the technique in considering long-term outcomes of action (related to side-effects and/or time delays) and can plan and act intelligently.

Technologies, Tools, and Resources used:

List of information needed by someone trying to reproduce this work.

- Swinburne Lecture Material
- Docs.python.org

Tasks undertaken:

Using the previous lab code as a base, we can implement advanced action selection functionality.

- The benefit of GOAP is that is consider side effects when selecting an action. To calculate the side effects, we need to create a function calcDiscontent.

```
-   def calcDiscontent(goals):
-       '''Calculate the total discontent based on the current goal values.'''
-       discontent = 0
-       for value in goals.values():
-           discontent += value * value
-       return discontent
```

- After we have our calculation function, we can create a choose action function that will choose the best available action based on the discontent values.

```
-   def choose_action():
-       assert len(goals) > 0, 'Need at least one goal'
-       assert len(actions) > 0, 'Need at least one action'
-
-       best_action = None
-       lowest_discontent = None
-
-       for key in actions.keys():
-           temp_goals = goals.copy()
-           apply_temp_action(key, temp_goals)
-           current_discontent = calcDiscontent(temp_goals)
-
-           if lowest_discontent is None or current_discontent <
    lowest_discontent:
-               lowest_discontent = current_discontent
-               best_action = key
-
-       return best_action
```

What we found out:
This implementation of GOAP shows that considering side effects and long-term outcomes of actions significantly improves the decision-making process in AI bots. By calculating the discontent values of goals after potential actions, the AI can choose the action that leads to the least overall discontent, leading to much smarter behaviour.

This spike relates to the following Unit Learning Outcomes (ULO):

- ULO 1: Discuss and implement software development techniques to support the creation of AI behaviour in games.
- The GOAP system is a key technique in AI behaviour development, supporting intelligent decision-making.
- ULO 2: Understand and utilize a variety of graph and path planning techniques.
- While GOAP itself isn't a path-planning technique, it’s use of a heuristic in order to choose the best “choice” is very similar to how a search algorithm works in a graph.
- ULO 4: Create agents that are capable of planning actions in order to achieve goals.
- The GOAP system directly addresses this outcome by creating a way for agents to plan and execute actions to minimise discontent.