

# CS4242 – Artificial Intelligence

## Decision Tree Project

### Goal

Utilize non-linear data structures and searching to generate correct agent response given an external/environmental event.

### Student Outcomes Addressed

2. Analyze and solving problems with search methods

### Details

Your task is to develop a program that reads in a behavior tree from an XML data file, build an in-memory representation of this behavior tree, and then provide a method by which the user can input an event to which the program will respond with the appropriate/correct response.

You can assume that all leaf nodes within the tree are the responses and all non-leaf nodes are keyed to events. It is possible for an event to have sub-trees. As an example, in the tree below (taken from GDC 2010 talk by Champandard, Dawe, and Hernandez-Cerpa), 'evade' is the only valid response to the 'incoming projectile' event; 'use computer' and 'patrol' are both valid responses to the input 'idle;' and there are five valid responses ('flee,' 'attack,' 'weapon 1,' 'weapon 2,' and 'weapon 3') to the input event 'combat.' For the input 'melee,' 'flee' and 'attack' are valid responses.



Your program can be written in any programming language you desire.

Note that if no match is found for the input event, the empty string should be returned.

We recommend you first verify that you can input the XML data file correctly by building this in memory and then displaying it out to the console/screen.

Hint: since this is a non-linear data structure, your solution will most definitely require recursion.

You must implement both a breadth-first and a depth-first search of the tree. Have your program output the results using both search techniques – it should give the response, then say “Item found in x steps with breadth-first, y steps for depth-first” for example.

## Grading

- 10% - overall structure of the program and style/clarity of code
- 20% - correctly inputting the XML file and building the tree in memory
- 30% - correct implementation of breadth-first search
- 30% - correct implementation of depth-first search
- 10% - correctly selecting a random response when more than one is possible

## Sample XML input

```
<root>
  <node behavior="Idle" response="">
    <node behavior="" response="Use Computer"/>
    <node behavior="" response="Patrol"/>
  </node>
  <node behavior="Incoming Projectile" response="">
    <node behavior="" response="Evade"/>
  </node>
  <node behavior="Combat" response="">
    <node behavior="Melee" response="">
      <node behavior="" response="Flee"/>
      <node behavior="" response="Attack"/>
    </node>
    <node behavior="Ranged" response="">
      <node behavior="" response="Weapon 1"/>
      <node behavior="" response="Weapon 2"/>
      <node behavior="" response="Weapon 3"/>
    </node>
  </node>
</root>
```

## Sample output (not necessarily correct, no cheating, right?)

```
-----
Behavior Tree Loaded...
```

```

behavior = ROOT
    behavior = Idle
        response = Use Computer
        response = Patrol
    behavior = Incoming Projectile
        response = Evade
    behavior = Combat
        behavior = Melee
            response = Flee
            response = Attack
        behavior = Ranged
            response = Weapon 1
            response = Weapon 2
            response = Weapon 3
-----
Event ('quit' to exit) : Combat
Response = Attack
Item found in 4 steps with BFS, 3 steps with DFS.
Event ('quit' to exit) : Combat
Response = Weapon 3
Item found in 5 steps with BFS, 5 steps with DFS.
Event ('quit' to exit) : Combat
Response = Weapon 1
Item found in 5 steps with BFS, 3 steps with DFS.
Event ('quit' to exit) : Combat
Response = Flee
Item found in 4 steps with BFS, 4 steps with DFS.
Event ('quit' to exit) : Combat
Response = Weapon 2
Item found in 4 steps with BFS, 4 steps with DFS.
Event ('quit' to exit) : Idle
Response = Patrol
Item found in 4 steps with BFS, 4 steps with DFS.
Event ('quit' to exit) : Idle
Response = Use Computer
Item found in 4 steps with BFS, 4 steps with DFS.
Event ('quit' to exit) : Idle
Response = Use Computer
Item found in 4 steps with BFS, 4 steps with DFS.
Event ('quit' to exit) : Idle
Response = Patrol
Item found in 4 steps with BFS, 4 steps with DFS.
Event ('quit' to exit) : quit

```