Agudo, Mikhail Joseph T.
Mamangun, Joseph
CS 171 First AI

1.0 Project Title

Attack of Arnold Schwarzenegger

2.0 Project Description

The AI uses the former deliverable in class with the same title. The agent acts in a fully observable, deterministic, sequential environment. The agent is named “John Connor” and the environment is a battle with a terminator named “Arnold”.

2.1 PEAS

2.1.1 Performance Measure

John Connor defeats Arnold, and John Connor does not die.

2.1.2 Environment

The terminator Arnold is the environment. Arnold can defend and fight back. Both John Connor and Arnold have hit points that will determine whether one or the other dies. John Connor has 1 hit point and Arnold has 2 hit points.

If John Connor attacks while Arnold is defending, Arnold will retaliate and attack John Connor. A successful attack will incur a loss of 1 hit point.

If John Connor does a successful attack against Arnold, Arnold will go into Defense Mode.

If John Connor is defending, Arnold’s Defense Mode will stop.

2.1.3 Actuators

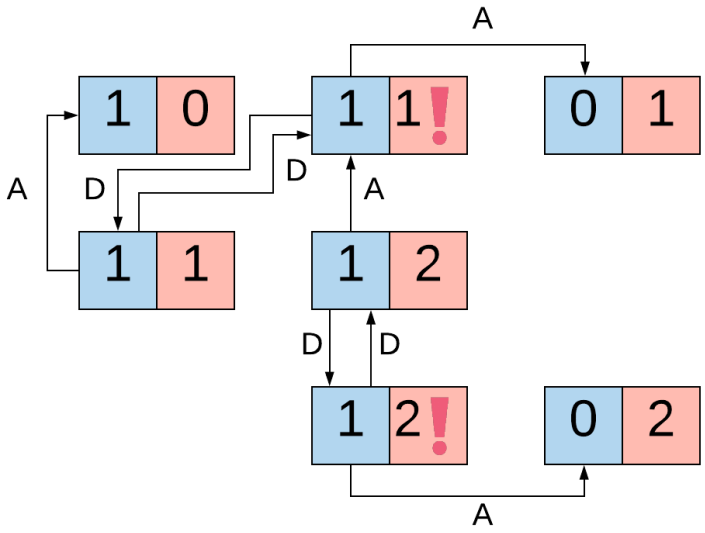
The AI can either attack or defend.

2.1.4 Sensors

John Connor is able to perceive the hit points of itself and Arnold. It can also perceive whether Arnold is defending from an attack or not.

2.2 State Transition Diagram

The State Transition Diagram has a total of 7 states. There is 1 Initial State, 1 Goal State and 2 Fail States.



3.0 Technical Description

The project uses Python 3.6.5, using no libraries, packages etc. There is a ui.py that uses the Tkinter library for the project but it is scrapped.

4.0 Code Description

The way the whole software works is: there is a main loop that allows the user to switch between Breadth-First Search and Iterative Deepening Search where the chosen option will start the tree traversal.

3.1 main.py

Contains the main loop for doing everything.

3.2 environment.py

Contains the code for Entity (John Connor and Arnold), and State which contains an object for John Connor and Arnold. There is a list that contains two strings “attack” and “defend” which will be important for ai.py.

3.3 utils.py

Contains code for Node, Queue and Bubble Sort. The Sort algorithm is used for A* Search and Greedy Best-First Search.

3.4 ai.py

Contains the algorithms for the two Uninformed Search Strategies and functions that will help in expanding a node, giving a state for the node and goal testing.

As of the Final Project, this now also contains the Informed Search Strategies: Greedy Best-First Search and A* Search. Within are also the evaluator functions for both search strategies.

3.5 ui.py

Contains the code for the UI. Currently scrapped.