**COMP2208 Assignment: Search Methods Report**

**Section 1: Approach**

**Depth First search:**

* Used last in first out so that we still get space complexity benefit of depth first search
* Created a method that returned one randomly chosen direction in which the agent should move

**Breadth First search:**

* Used in a first in first out list that helps schedule the nodes to expand allowing to fulfill the breadth first function.
* To combat the likelihood of going into a loop by making the agent follow preset movement directions I have created a method that returns the directions the agent should be moved in a randomized order.

**General:**

* When an algorithm tries to move the agent in a particular direction it returns true or false based on whether the move will have an affect on the agents position. E.g if the agent is touching the left wall and the agent is told to move left, the method for moving will return false. This allows the algorithm to only add nodes to the tree (list) that actually changed the position of the agent, making it impossible for the agent to stay stuck in one position on the board for more than one node in a row.
* Keeping time complexity of the algorithm in mind I did not want the algorithm to have to loop through the 2D array every time a node is examined to check if it is in the final state. Instead I have created a variable for the position of blocks A, B, C. Like this the algorithm just have to compare the variables, reducing the time complexity of the algorithms.

**Section 2: Evidence of Search**

For space (due to 6 page limit) and time (time to finish search) I have stuck to 3x3 size boards. The program will automatically remove the ‘c’ block from the search space.

**Depth First search:**

Total moves (this run): 104

1 2 3 4 5 6 7 8 9

ppp

ppp

asb

ppp

ppb

aps

ppp

ppb

asp

ppp

ppb

sap

ppp

ppb

asp

ppp

ppb

aps

ppp

pps

apb

ppp

psp

apb

ppp

ppp

abs

Final moves list:

[l, u, r, d, l, l, r, r, l, l, u, u, d, u, r, r, l, d, d, r, l, u, d, u, r, u, d, d, u, l, u, d, r, u, l, r, d, l, u, d, u, l, r, l, d, r, r, d, u, d, l, r, l, l, u, r, d, r, l, l, r, l, u, d, u, r, l, r, u, r, d, d, l, u, d, r, u, l, d, l, r, r, u, l, u, d, l, u, d, d, r, l, r, l, u, u, d, r, d, u, d, u, d, l]

**Breadth First search:**

Total moves (this run): 528

1 2 3 4 5 6 7 8 9

ppp

pps

abp

ppp

ppp

abs

ppp

ppp

sab

ppp

psp

apb

pps

ppp

abp

ppp

psp

abp

ppp

ppp

abs

ppp

ppp

asb

ppp

ppp

abs