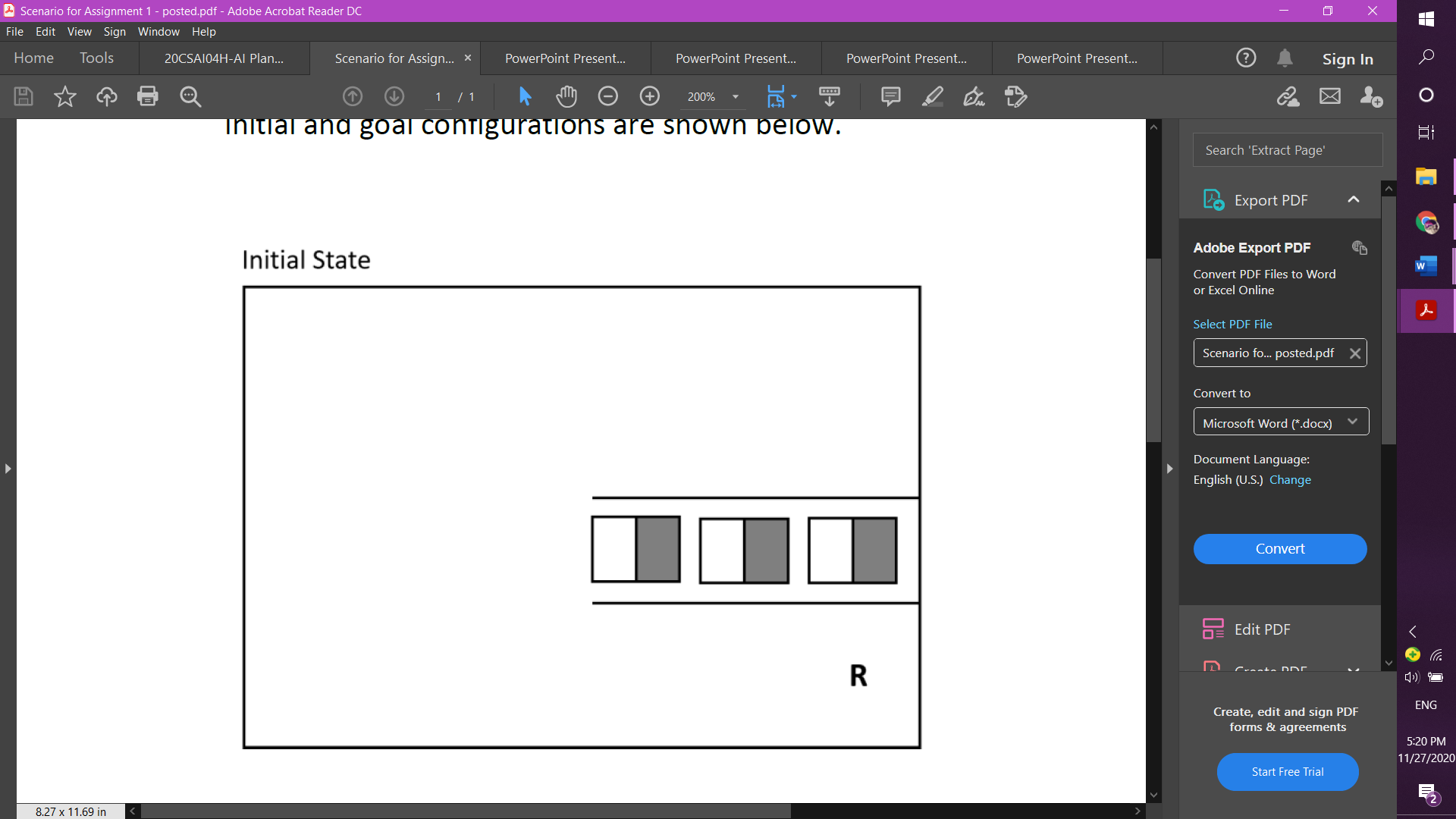
**Assignment 1 AI planning for robot systems**



Destination1\_way

Destination2\_way

Destination3\_way

nonBoxWay

R\_loc

Box1way

Box1 Box2 Box3

Destination1

Destination2

Destination3

nonBoxWay

**The starting point:** R\_loc.

**The ending point:** returning to R\_loc.

**The goal:** moving the 3 boxes and reordering them in the provided destination.

**The scenario:**

**First,** the robot ‘R’ will move using directions (north, south, east, west), it will move through non\_box\_way to the west then to the south then to the east until it reaches Box1way, then it will pick up the box1, after that, it will move to the west again until it finds destination 1, the robot won’t be able to put down at destination1 because it will be standing there, so it will put down box 1 and turn around it by moving north then east then west, after that, it will pick up box 1 again then move west then put it down, after reaching the first achievement by putting box 1 at destination 1 it should go and get the other boxes. **Second,** the robot will move east until it reaches box2 way then it will pick up box 2 then move west then north until it finds destination 2, the robot won’t be able to put down at destination2 because it will be standing there, so it will put down box 2 and turn around it by moving north then east then west, after that, it will pick up box 2 again then move west then put it down at destination2, after reaching the second achievement by putting box 2 at destination 2 it should go and get the other box. **Third,** the robot will move east then south then east until it reaches box3 way then it will pick up box 3 then move west then north then west until it finds destination 2, the robot won’t be able to put down at destination3 because the robot will be standing there, so it will put down box 3 and turn around it by moving north then east then west, after that, it will pick up box 3 again then move west then put it down at destination3, after reaching the third achievement by putting box 3 at destination 3 the robot should turn back to the starting point so it will move east then south then east until it reaches the starting point R\_loc after that the goal will be accomplished.

**Environment:**

[objects[R],[Box1],[Box2],[Box3]]

[locations[R\_loc],[nonBoxway],[Box1way],[Box2way],[Box3way],[ Destination1\_way],[ Destination2\_way],[Destination3\_way],[Destination1],[Destination2],[Destination3]]

[connections [R],[Box1way,Box1,Box2way,Box2,Box3way,Box3],[ Destination1\_way, Destination2\_way, Destination3\_way, Destination1,Destination2,Destination3]].

**Goal directed/ Assembly planning:**

To reach the goal by moving object (B) from location (X) to location (D) by the object (R)

1.move\_to(X) 2.pick\_up(B) 3.put\_down(B) 4. move\_object\_to(B,D) 5.move\_around(B)

|  |  |
| --- | --- |
| **Predicates** | At(X), holding(B), holderEmpty, BoxIsThere(B), R\_arms\_to\_right, R\_arms\_to\_left |
| **Move\_to(X,Y)**  **X:** the nonBoxWay or BoxWay or R\_loc or Destination\_way or Destination2\_way or Destination3\_way.  **Y:** the nonBoxWay or BoxWay or R\_loc or Destination1\_way or Destination2\_way or Destination3\_way. | **Preconditions:** At(X)  **Delete:** At(X)  **Add:** At(Y) |
| **Pick\_up(B)**  **B:** Box1 or Box2 or Box3 | **Preconditions:** holderEmpty, BoxIsThere(B)  **Delete:** holderEmpty, BoxIsThere(B)  **Add:** holding(B) |
| **Put\_down(B)**  **B:** Box1 or Box2 or Box3 | **Preconditions:** holding(B)  **Delete:** holding(B)  **Add:** holderEmpty |
| **Rotate\_robot\_arms\_left()** | **Preconditions:** R\_arms\_to\_right  **Delete:** R\_arms\_to\_right  **Add:** rotate\_arms\_to\_left |
| **Rotate\_robot\_arms\_right()** | **Preconditions:** R\_arms\_to\_left  **Delete:** R\_arms\_to\_left  **Add:** rotate\_arms\_to\_right |

**The problem represented as a graph:**

D3 D3way

D2 D2way

B1way B1 B2 B3

D1 D1way

R

**Plan:**

Move\_to(start,Box1way), Pick\_up(Box1), Move\_to(Box1way ,Destination1\_way), Rotate\_robot\_arms\_left(), Put\_down(Box1), Move\_to(Destination1\_way ,Box1), Rotate\_robot\_arms\_right(), Pick\_up(Box2), Move\_to(Box1,Destination2\_way), Rotate\_robot\_arms\_left(), Put\_down(Box2), Move\_to(Destination2\_way ,Box2), Rotate\_robot\_arms\_right(),Pick\_up(Box3), Move\_to(Box2,Destination3\_way), Rotate\_robot\_arms\_left(), Put\_down(Box3), Move\_to(Destination3\_way ,R\_loc).

**Planning on 2D Grid:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| (3,0)  Destina-  tion3 | (3,1)  Destination3  \_way | (3,2) | (3,3) | (3,4) | (3,5) |
| (2,0)  Destinat-  Ion2 | (2,1)  Destination2  \_way | (2,2) | (2,3) | (2,4) | (2,5) |
| (1,0)  Destinat-  Ion1 | (1,1)  Destination1  \_way | (1,2)  Box1way | (1,3)  Box1 | (1,4)  Box2 | (1,5)  Box3 |
| (0,0) | (0,1) | (0,2) | (0,3) | (0,4) | (0,5)  R |

**Note:** cells with gray highlight are objects and cells with gray borders are locations.

**The possibility is:** both orthogonal and diagonal at the same time.

**The used algorithm to find the shortest path:**

Breadth first search (BFS) because it will search level by level and we are moving both orthogonal and diagonal in the grid to move and search quickly with out time consuming to reach the goal.

The starting and ending points are at R 🡺 (0,5)

**The shortest path:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| (3,0)  Destina-  tion3 | (3,1)  Destination3  \_way | (3,2) | (3,3) | (3,4) | (3,5) |
| (2,0)  Destinat-  Ion2 | (2,1)  Destination2  \_way | (2,2) | (2,3) | (2,4) | (2,5) |
| (1,0)  Destinat-  Ion1 | (1,1)  Destination1  \_way | (1,2)  Box1way | (1,3)  Box1 | (1,4)  Box2 | (1,5)  Box3 |
| (0,0) | (0,1) | (0,2) | (0,3) | (0,4) | (0,5)  R |

**Resources:**

<https://www.youtube.com/watch?v=KiCBXu4P-2Y&ab_channel=WilliamFiset>  
<https://www.youtube.com/watch?v=x6cGmE0XpY8>

**Name:** Rawan Abdellah