Valet Assignment Report

Hybrid A* algorithm on dynamic sampling-based graph generation was used to solve the valet problem. Here only the nodes of the graph that are relevant are created and explored. This saves us a lot of time in generating a static graph. Different cost for different action was used to penalize reversing of the car and turning off the car.

The Following algorithm has been used to find motion plans in all three problems:

Path Finding Algorithm Pseudo Code:

```
Step 1: Initialize root_node and goal_state
Step 2: add root_node in the queue
Step 3: while queue is not empty:
            Current_node = sorted(queue).pop()
            If current_node is at goal state:
                Goal_node = current_node
                Stop
            Child_nodes = current_node.find_child()
            For all child nodes:
                If child_node is not visited and child_node is not colliding with obstacles:
                    Add to queue with heuristic cost
Step4: path = []
Step5:node = goal_node
Step5: while node.parent != root_node:
            path.append(node)
            Node = node.parent
Step6:path = reverse(path)
```

Report:
Each problem(Delivery Bot, Car, and Truck) has a folder named after itself. It contains a video of the path following, the image of the path, the image of the vehicle, car.py, c_space_generator.py, a_star_palnner.py.

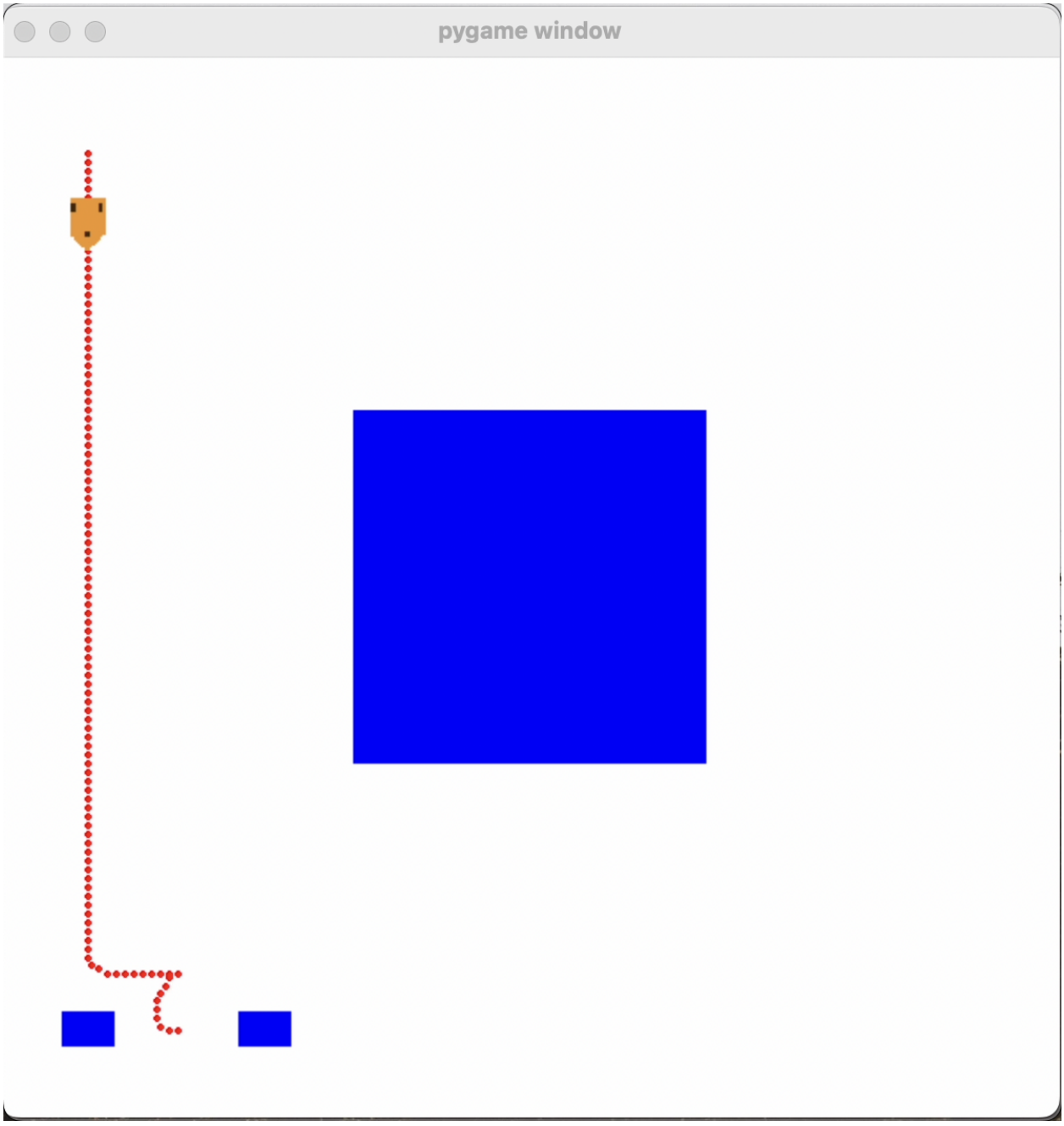
Car.py file contains a class CAR. here I have implemented pygame draw function, next_state calculations based on the kinematics, and other necessary functions.

C_space_generator.py contains a class node that is used to store the nodes of the graph.
Here each object of this class will be a node of the graph. It will store its state, its parent object for path retracing, and its child_nodes. It has a function to find child nodes of the current node. I have used find_child() to extend the node into four nodes: vehicle move forward, move backward, turn left, turn right. This function is also used to give cost to the actions.

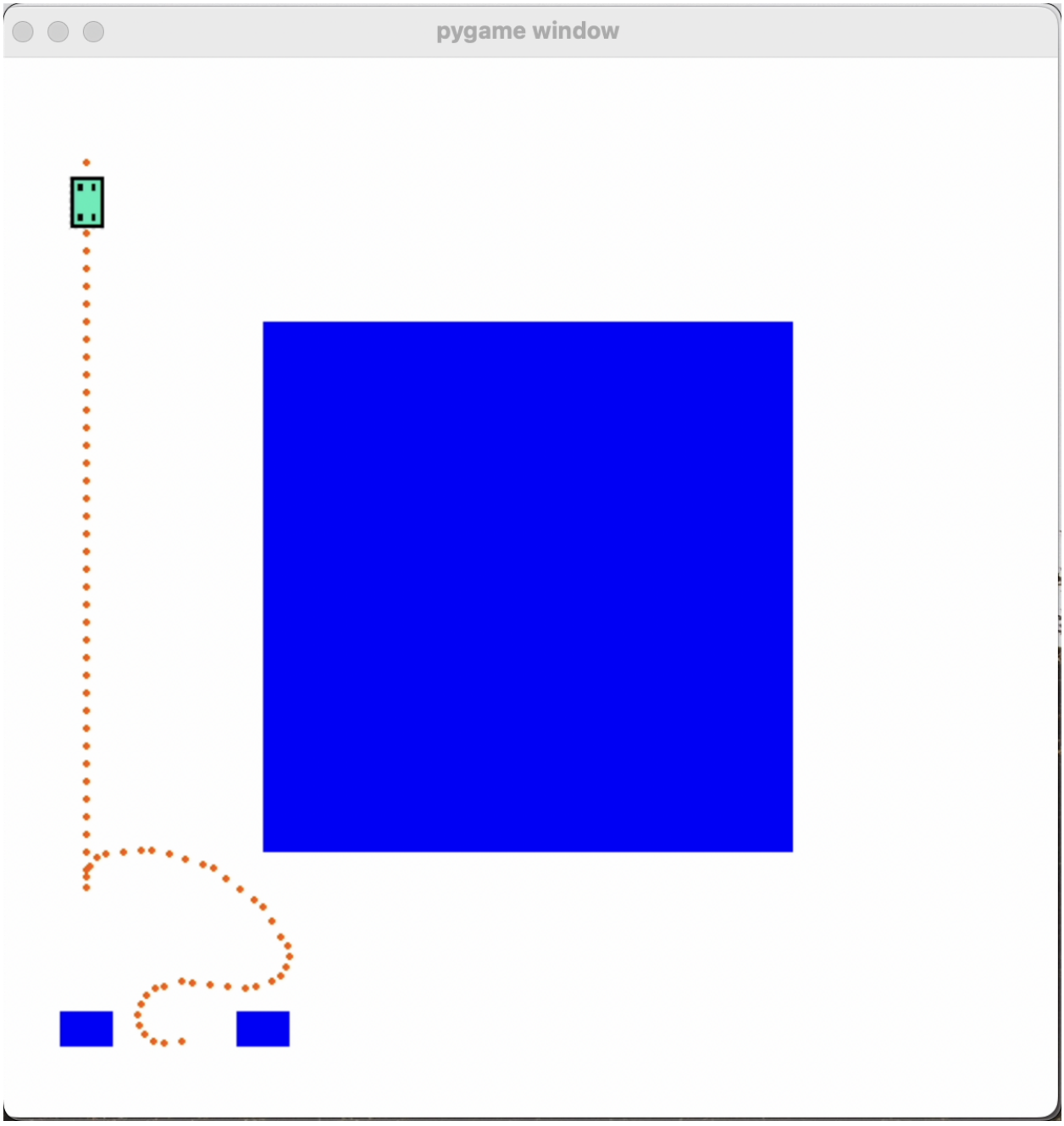
A_star_planner.py contains the main path planning function and this is the file that needs to be run to the solution of the code.

Results:

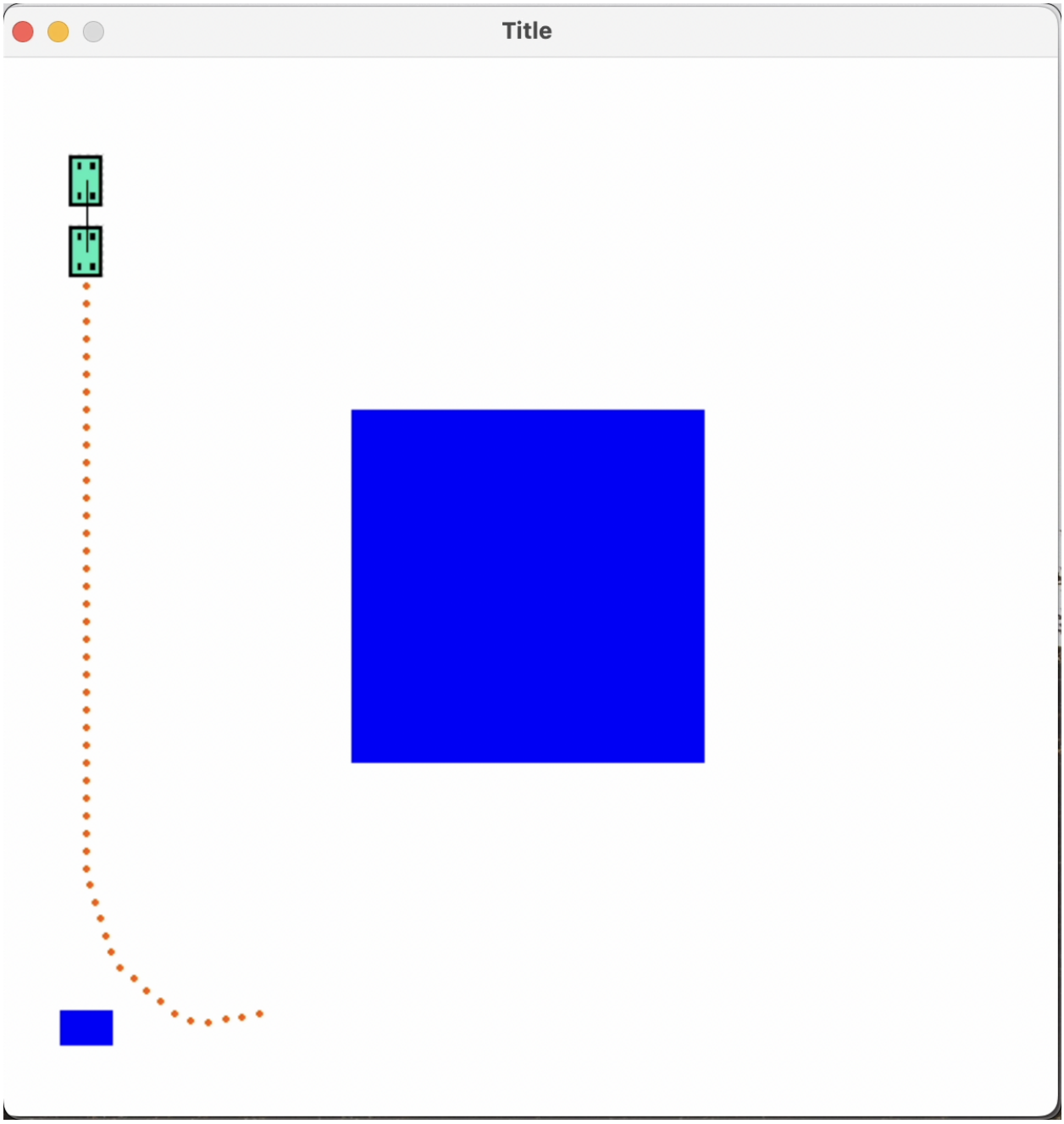
Videos of the path following can be found in individual folders for each problem along with source code.



Delivery Bot path



Car path



Truck path