Exercises 1-2: Sampling-Based Motion Planning (ex.py)

1. Uncomment problem="1". The display shows a circular robot moving in a square domain amongst a circular obstacle. You can press "p" to plan for 100 iterations, and press [space] to plan for a single iteration. When a plan is found, it is displayed.

    a. Correctly implement the feasibility test in CircleObstacleCSpace so that the robot does not actually collide with the boundaries of the region and the obstacles.
    b. Investigate how performance differs between the 'prm' and 'rrt' settings to the motion planner. These settings are specified in the MotionPlanning.setOptions calls in the CSpaceObstacleProgram initializer. Tune the connectionThreshold and perturbationThreshold parameters and investigate how they affect planning performance and the shape of the solution path. Discuss the results of your testing. (Include figures or snapshots if appropriate)

2. Uncomment problem="2". The display shows a rigid bar robot with configuration space SE(2). The RigidBarCSpace implementation is incomplete: first, it does not correctly test for collisions along the entire bar; second, it does not correctly interpolate and return a distance metric on SE(2).
    a. Implement the C-space correctly. For verification, show a snapshot of the output path.
    b. Discuss the weighting of the angular component. How would it affect a PRM planner? How would it affect an RRT planner? Rotate the start configuration to have orientation π. Perform some experiments with a high weight (10), a moderate weight (0.1) and a low weight (0). What performs the best? Why do you think this is so?