Your task is to build a controller for turtlebot, which takes a goal point as input, plans a path through a field of static obstacles and then publishes appropriate values to cmd vel. We have provided the world file as well as the corresponding launch file required.

We have created the world file consisting of static obstacles. All obstacles are 10m tall cylinders with a 0.25m radius. All cylinders have been located in grid at locations { (0, 1.5), (0, 3), (0, 4.5), (1.5, 0), (1.5, 1.5), (1.5, 3), (1.5, 4.5) … (4.5, 0), (4.5, 1.5), (4.5, 3), (4.5, 4.5) } with the bot initialized at (0, 0). The goal point is (6, 6). In your path planning algorithm, you can take this list as input. When you want to check if a point is inside an obstacle (something commonly required in sampling-based algorithms), you can simply check if the point is within a certain distance of any of the provided centre points. Also, don't forget to include the size of the robot in this calculation.

2.2 Required Structure:

1. Obstacle detector node, which publishes a fixed list of obstacles. Here you will be publishing the list of the centre point we have provided.
2. Planner node subscribes to obstacles and publishes a path. You can take any two appropriate points in the world for start and end.
3. Controller node which subscribes to the path and publishes cmd_vel. This controller will traverse the path one point at a time using PID.

For convenience, you should write a launch file so that you won't have to run all three nodes each time.
You will need to demonstrate the robot successfully navigating the given obstacle field without knocking into any obstacles. You can choose any algorithm from the following:

1. RRT*
2. PRM+A*
3. Artificial Potential FIeld

All this might seem a little intimidating. The key thing, however, is to break it up into manageable chunks.