

```
1 # For sampling: The code is used to generate the data for creation of dataset for
model training. It reads the generated maps and provides the result of various
sampling methods in those maps, giving output in the form of a text file
```

```
1 # For testing: This code reads the test map and tests the different sampling
methods in different segments of the map. The map file and the information
regarding sampling strategies for different segments is taken as input. It generates
a map with non uniform sampling and finds a path for the robot
```

```
1 # Install Required Libraries:
```

```
1 # The code is mainly divided into two parts:
```

```
In [ ]: 1 #1). Sampling and Testing
2 #2). Training and using the Learning model
```

```
1 # Sampling and Testing:
```

```
In [1]: 1 #1).For sampling:
2 #Changes that are needed:
3 1. Update the sampling method that is needed in the file main_New.py
4 2. Update the location of map files.
5 3. Update the start and goal locations.
6 4. In the file Planner.py, ensure that in the 'plan' method of planner class, self.method.plan is called.
7 5. In the file PRM_New.py, update the value in 'sample' method of PRM class. The 'value' will be step size
8 i.e. It will take 'value' number of samples and check for a path.
9 If path is not found, it will repeat the same process.
10 6. Run the file main_New.py.
```

```
In [ ]: 1 #2).For Testing:
2 Changes the are needed:
3 1. Update the sampling methods in the list "method" in the file 'Test_Main.py'. Comment in the code instructs
4 how to update.
5 2. Update the location of map file.
6 3. Update the start and goal locations.
7 4. In the file Planner.py, ensure that in the 'plan' method of planner class, self.method.test_plan is called.
8 4. Run the file Test_Main.py.
```

```
In [ ]: 1 #Use 'CNN_Obstacle_Classification.py' to classify the map segmented image
```

```
In [ ]: 1 #For any clarifications, please feel free to reach out to 'dkotadia@wpi.edu'.
```