



Name _____

Class _____

Date _____

What this lesson is about

Throwing darts at the free space instead of enumerating it, and what probabilistic completeness does and does not promise.

- The RRT
- What it promises
- Randomness has consequences

Questions

6 marks in all

1. A robot on the 200 by 200 cm mat plans over (x, y, heading) with 10 cm cells and 10 degree heading steps. How many cells does the grid have? [1 mark]

2. There is no route from the start to the goal. What does a plain RRT do? [1 mark]
 - ☐ A It keeps sampling until it is stopped, and can never report that no route exists
 - ☐ B It reports no route after it has sampled every cell
 - ☐ C It returns the node nearest to the goal
 - ☐ D It returns a route through the narrowest wall
3. Why does an RRT grow outwards into unexplored space without being told to? [1 mark]
 - ☐ A A sample is extended from the nearest node, and nodes on the edge of the tree have the largest Voronoi regions
 - ☐ B The goal bias pulls the tree outwards
 - ☐ C New nodes are only accepted if they are further from the start
 - ☐ D The step size grows as the tree gets bigger
4. What does this program print? [1 mark]

```
def free(x, y):  
    return not (60.0 <= x <= 80.0 and 0.0 <= y <= 100.0)  
  
a, b = (50.0, 50.0), (90.0, 50.0)  
print(free(*a), free(*b))  
n = 20  
points = [(a[0] + (b[0] - a[0]) * k / n, a[1] + (b[1] - a[1]) * k / n) for k in range(n + 1)]  
print(all(free(x, y) for x, y in points))
```


5. The goal bias is set so the RRT samples the goal most of the time. What happens? [1 mark]
- ☐ A It behaves like greedy best first and gets stuck against a wall between it and the goal
 - ☐ B It becomes optimal
 - ☐ C It explores the whole space faster
 - ☐ D It becomes deterministic and always finds the same route
6. Which of these statements are true? [1 mark]

Tick every answer that is true.

- ☐ A More samples do not make a plain RRT converge to the optimal path
- ☐ B RRT* rewires nearby nodes and is asymptotically optimal
- ☐ C A PRM suits a fixed map with many different start and goal queries
- ☐ D Shortcutting should test lines of sight against the raw, uninflated obstacles

The task: grow a tree into the free space

Build an RRT from (30, 30) to (170, 170) through the same two walls, inflated by 10 cm. Print `nodes:`, how many nodes the tree had when it reached the goal, and `path:`, the length in centimetres of the route through the tree.

```
from bugbot import *
import math
import random
connect()

INFLATE = 10.0
STEP = 12.0
BIAS = 0.1
START = (30.0, 30.0)
GOAL = (170.0, 170.0)
WALLS = [(70.0, 0.0, 8.0, 115.0), (125.0, 85.0, 8.0, 115.0)]
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/u9-6-sampling-and-rrt/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Shortcut the finished path and print the length before and after. How much of the loss does it recover?
2. Run with the goal bias at 0, 0.05, 0.5 and 1.0, ten times each, and report the mean number of nodes. Explain both ends of the curve.
3. Take the edge check out, so only the new point is tested. Find a seed where the path goes through a wall.