

# Rapidly Exploring Random Trees - Assignment 4

Himanshu Kihisor Choubey  
190376

April 2022

## 1 Information regarding RRT code

1. Obstacles are marked by black boundaries.
2. Obstacles are dynamic (made use mouse).
3. Start point is yellow.
4. End point is red.
5. Length of each shot node is 10, the same as `length` of boundary of obstacles.
6. The above lengths were chosen to easily define obstacles and shot node does not reach inside the obstacle

## 2 Python Code using Pygame Library for RRT

```
import sys,pygame

import os

import time

import math

import random

from random import randint as ri

pygame.font.init()

pygame.init()

WIDTH, HEIGHT = 900, 500

gui_font = pygame.font.Font(None, 30)

WIN = pygame.display.set_mode((WIDTH, HEIGHT))
```

```

pygame.display.set_caption("RRT")

BORDER = pygame.Rect(445, 0, 10, HEIGHT)

#COLORS DEFINED HERE

WHITE = (255,255,255)

BLACK = (0,0,0)

BLUE = (0,0,255)

RED = (255,0,0)

YELLOW = (255, 255, 0)

GREY = (128,128,128)

FPS = 30

WIN.fill(WHITE)

class Button:
    def __init__(self, text, colour,x,y,width,height,level):
        self.colour = colour
        self.x = x
        self.y = y
        self.width = width
        self.height = height
        self.level = level
        self.text_surf = gui_font.render(text, True, BLACK)
        self.text_rect = self.text_surf.get_rect(center = (x+width//2, y+height//2))
        self.top_rect = pygame.Rect((x,y), (width, height))

    def create(self, WIN):
        pygame.draw.rect(WIN, self.colour, [self.x, self.y,self.width ,self.height])
        WIN.blit(self.text_surf, self.text_rect)

#check if new node inside domain

def valid(curr_x,curr_y,end_pos):
    if(end_pos[0]>=825 or end_pos[1]>=400 or end_pos[0]<=25 or end_pos[1]<=25):
        return 1

```

```
#shoot node
```

```
def shoot_node(d, ang, curr_x, curr_y):

    end_pos = ((curr_x+d*math.cos(ang*math.pi/180)), (curr_y+d*math.sin(ang*math.pi/180))
    if(valid(curr_x,curr_y,end_pos)):
        return (curr_x, curr_y)

    end_pos = ((curr_x+d*math.cos(ang*math.pi/180)), (curr_y+d*math.sin(ang*math.pi/180))
    end_pos = (math.floor(end_pos[0]), math.floor(end_pos[1]))

    # print(grey)
    # print(end_pos)

    for i in range(10):
        for j in range(10):
            if (end_pos[0]+i, end_pos[1]+j) in grey:
                return (curr_x, curr_y)

    if not (valid(curr_x,curr_y,end_pos)):
        pygame.draw.line(WIN, BLUE, (curr_x, curr_y), end_pos, 2)
    pygame.display.update()
    return end_pos


def solve_rrt(curr_x, curr_y, end_x, end_y):
    d = 10
    ang = ri(0,360)
    while(not (curr_x>=end_x and curr_y>=end_y and curr_x-10<=end_x and curr_y-10<=end_y):

        jj = shoot_node(d,ang, curr_x, curr_y)
        ang = ri(0,360)
        curr_x = jj[0]
        curr_y = jj[1]

    # run = False


def draw_window(b_color):
    pygame.draw.rect(WIN,BLACK,(25,25,825,400),5)
```

```

        # pygame.draw.rect(WIN, b_color, (25, 450, 25, 25))
        pygame.display.update()

    grey = {}
    yellow = {}
    red = {}

def main():
    level = 1
    run = True
    clock = pygame.time.Clock()
    press = False
    b_color = GREY

    while(run):
        clock.tick(FPS)

        if(level == 1):
            b_color = GREY
        if(level == 2):
            b_color = YELLOW
        if level==3:
            b_color = RED
        if level ==4:
            solve_rrt(start_x, start_y, end_x, end_y)
            level+=1
            # pygame.quit() #CHANGES HEREEEE!!!!!!!!!!!!!!

        mouse_pos = pygame.mouse.get_pos()

        if mouse_pos[0]>=25 and mouse_pos[1]>=435 and mouse_pos[0]<=25+100 and mouse
            if pygame.mouse.get_pressed()[0]:
                press = True
            else:
                if(press == True):
                    level+=1
                    press = False
        B1 = Button('Next', b_color, 25, 435, 100, 50, level)
        B1.create(WIN)

    for event in pygame.event.get():
        if event.type == pygame.QUIT:

```

```

        run = False
        pygame.quit()
    if run==False:
        pygame.quit()
        break

    m = pygame.mouse.get_pressed()
    x,y = pygame.mouse.get_pos()
    # print(x,y)

    if(m[0]==1 and x>=25 and y>=25 and x<=840 and y<=415):
        if(b_color==GREY):
            b_color = BLACK

        pygame.draw.rect(WIN, b_color, (x,y,10,10))

        if(b_color == BLACK):
            b_color = GREY
        if(b_color==GREY):
            grey[(x,y)]=1
            grey[(x+10,y+10)] = 1
        if(b_color==YELLOW):
            start_x = x
            start_y = y
        if(b_color==RED):
            end_x = x
            end_y = y

    if event.type == pygame.MOUSEBUTTONUP:
        press == False

    draw_window(b_color)

    main()

if __name__=="__main__":
    main()

```

### 3 RRT path for the obstacles

