# Task 1

In this task, we will translate the code into OOP.

For convenience, we also create a helper, just to make sure the code less clusted.

Here, we create a .py name robot\_helper, and

def initialise(window):  
 window.resizable(False,False)  
 canvas = tk.Canvas(window,width=1000,height=1000)  
 canvas.pack()  
 return canvas

and now, to call this,

from robot\_helper import initialize

# Task 2

In the second task, you should create a class dirt

and since we do might not change this code much, let put this code under a new file.mmm, lets name it passive component

class Dirt:  
 def \_\_init\_\_(self,namep):  
 self.centreX = random.randint(100,900)  
 self.centreY = random.randint(100,900)  
 self.name = namep  
  
 def draw(self,canvas):  
 dirt\_constant=4  
 body = canvas.create\_oval(self.centreX-dirt\_constant,self.centreY-dirt\_constant, \  
 self.centreX+dirt\_constant,self.centreY+dirt\_constant, \  
 fill="grey",tags=self.name)  
  
 def getLocation(self):  
 return self.centreX, self.centreY

and to call the class, we can use

from passive\_component import Dirt

# Task 3

Next, lets create a class for the dirt counting

class Counter:  
 def \_\_init\_\_(self,canvas):  
 self.dirtCollected = 0  
 self.canvas = canvas  
 self.canvas.create\_text(70,50,text="Dirt collected: "+str(self.dirtCollected),tags="counter")  
   
 def itemCollected(self, canvas):  
 self.dirtCollected +=1  
 self.canvas.itemconfigure("counter",text="Dirt collected: "+str(self.dirtCollected))

paste the code the passive component,

remember to add import

and call it by replacing

count = Counter(canvas)

under the function

def register(canvas):

when you run the code, the counter should be there