**Simulating a container terminal using SimPy**

SANTHOSH A

[asanthosh2502@gmail.com](mailto:asanthosh2502@gmail.com)

UNDERSTANDING:

Simpy is a open source python frame work used for simulation.Using Simpy I have simulated the container terminal with help of resource from simpy,generators from python,timeout event from simpy,process from simpy and basic oops concepts.

1. 2 berths are available.At a time only 2 vessels can occupy it. Other vessels have to wait until a berth becomes free.

2.2 Quay Cranes are available.A vessel should use only one crane.

2.1.It is achieved using aloocating and releasing crane resource manually.

3. 3 Trucks are available.Crane can use any of three trucks.

The fixed variables such as Berth,Cranes,Trucks are placed inside a class.In case of changes , we need to change in one place only .This helps to achieve integrity.

The Truck and cranes inherit Equipment class because they both need request and release methods.Further they contain extra methods specific to each .

In case of Vessels containing different amount of containers it can also be changes easily.

GithubLink: <https://github.com/santhosha123/Container_Terminal_Stimulator.git>

DriveLink: https://drive.google.com/drive/folders/1L2SOOQ2tVqPpZXdy8BsBVyBaESbfcjBD?usp=drive\_link

Code:

import simpy

import random

class ContainerTerminal:

no\_of\_crane=2

no\_of\_truck=3

no\_of\_berth=2

no\_of\_container = 150

class Vessel:

# Similar to constructor overloading in Java. If container count is passed then it uses the passed value or else default value

def \_\_init\_\_(self, vessel\_id, container\_count=ContainerTerminal.no\_of\_container):

self.vessel\_id=vessel\_id

self.container\_count=container\_count

class Equipment:

def \_\_init\_\_(self, env, capacity):

self.env=env

self.resource=simpy.Resource(env, capacity)

def request(self):

return self.resource.request()

def release(self, request):

self.resource.release(request)

class Crane(Equipment):

def \_\_init\_\_(self, env, capacity):

super().\_\_init\_\_(env, capacity)

def lift\_container(self, vessel\_id, container\_id):

print(f"Time {self.env.now:.2f}: Crane starts lifting container {container\_id} from vessel {vessel\_id}.")

yield self.env.timeout(3) # Time taken by a crane to lift one container is 3 minutes

class Truck(Equipment):

def \_\_init\_\_(self, env, capacity):

super().\_\_init\_\_(env, capacity)

def transport\_container(self, vessel\_id, container\_id):

print(f"Time {self.env.now:.2f}: Truck starts transporting container {container\_id} from vessel {vessel\_id} to yard block.")

yield self.env.timeout(6) # Time for truck to transport container and return

print(f"Time {self.env.now:.2f}: Truck returned after dropping container {container\_id} from vessel {vessel\_id}.")

class ContainerTerminalWorking:

def \_\_init\_\_(self, env):

self.env=env

self.berths=simpy.Resource(env, ContainerTerminal.no\_of\_berth)

self.cranes=Crane(env, ContainerTerminal.no\_of\_crane)

self.trucks=Truck(env, ContainerTerminal.no\_of\_truck)

def unload\_vessel(self, vessel):

print(f"Time {self.env.now:.2f}: Vessel {vessel.vessel\_id} enters waiting queue.")

# Requesting berth for the vessel

with self.berths.request() as berth:

yield berth

print(f"Time {self.env.now:.2f}: Vessel {vessel.vessel\_id} berths.")

# Requesting cranes if available and ensuring that a single crane is allocated to entire containers from a vessel by releasing the resource manually

crane\_request = self.cranes.request()

yield crane\_request

try:

for i in range(vessel.container\_count):

yield self.env.process(self.cranes.lift\_container(vessel.vessel\_id, i + 1))

self.env.process(self.move\_container(vessel.vessel\_id, i + 1))

# Time taken by crane to look for a free truck

yield self.env.timeout(1)

finally:

# Releasing the Crane resource manually

self.cranes.release(crane\_request)

print(f"Time {self.env.now:.2f}: Vessel {vessel.vessel\_id} finishes unloading and leaves.")

self.berths.release(berth)

def move\_container(self, vessel\_id, container\_id):

with self.trucks.request() as truck\_request:

yield truck\_request

yield self.env.process(self.trucks.transport\_container(vessel\_id, container\_id))

def vessel\_arrival(self):

vessel\_id=1

while True:

vessel=Vessel(vessel\_id)

self.env.process(self.unload\_vessel(vessel))

vessel\_id+=1

inter\_arrival\_time=random.expovariate(1 / 300) # Average of 5 hours (300 minutes)

yield self.env.timeout(inter\_arrival\_time)

# Initialize environment

env=simpy.Environment()

# Create the container\_terminal\_working obj

simulation = ContainerTerminalWorking(env)

# Start vessel arrival process

env.process(simulation.vessel\_arrival())

# Run the simulation for 1440 minutes (24 hours)

env.run(until=1440)

Output:

Time 0.00: Vessel 1 enters waiting queue.

Time 0.00: Vessel 1 berths.

Time 0.00: Crane starts lifting container 1 from vessel 1.

Time 3.00: Truck starts transporting container 1 from vessel 1 to yard block.

Time 4.00: Crane starts lifting container 2 from vessel 1.

Time 7.00: Truck starts transporting container 2 from vessel 1 to yard block.

Time 8.00: Crane starts lifting container 3 from vessel 1.

Time 9.00: Truck returned after dropping container 1 from vessel 1.

Time 11.00: Truck starts transporting container 3 from vessel 1 to yard block.

Time 12.00: Crane starts lifting container 4 from vessel 1.

Time 13.00: Truck returned after dropping container 2 from vessel 1.

Time 15.00: Truck starts transporting container 4 from vessel 1 to yard block.

Time 16.00: Crane starts lifting container 5 from vessel 1.

Time 17.00: Truck returned after dropping container 3 from vessel 1.

Time 19.00: Truck starts transporting container 5 from vessel 1 to yard block.

Time 20.00: Crane starts lifting container 6 from vessel 1.

Time 21.00: Truck returned after dropping container 4 from vessel 1.

Time 23.00: Truck starts transporting container 6 from vessel 1 to yard block.

Time 24.00: Crane starts lifting container 7 from vessel 1.

Time 25.00: Truck returned after dropping container 5 from vessel 1.

Time 27.00: Truck starts transporting container 7 from vessel 1 to yard block.

If possible after examining could you send me feedback and any other ideas to implement the projects.

Thanking You.