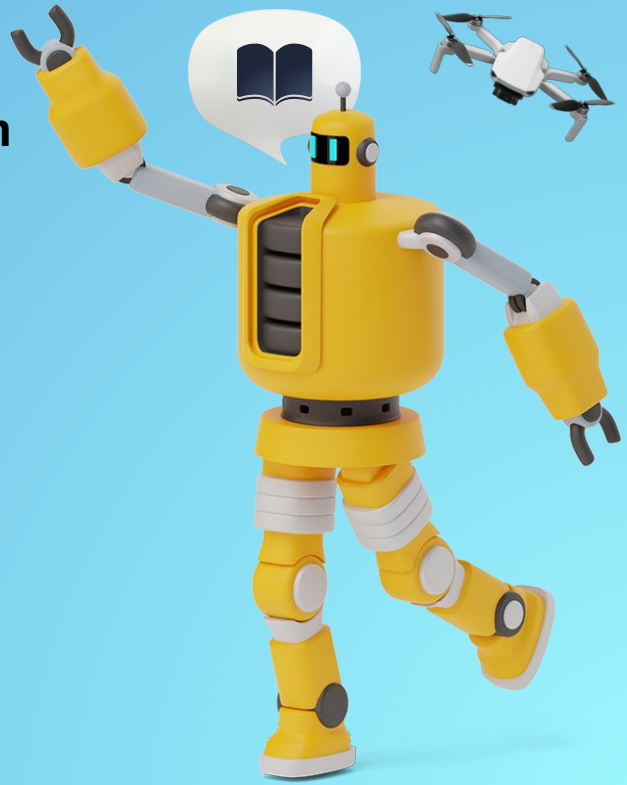
RECHARGING:

GROUP 22'S THIRD ITERATION PROJECT DELIVERABLE

PRESENTED BY: AUNYA MUKHERJEE, DUSTIN ZHANG, LIN HUYNH, AND WILEY BUI

INTRODUCTION AND MOTIVATION

- **Introduction to the Package Delivery Simulation**
  - **Customer**
  - **Package**
  - **Drone**
  - **Robot**
- **Motivation for Recharging**



BIG PICTURE

How does this fit in to what we already have?

IMPLEMENTATION

How did we do it?

IMPLEMENTATION

Design process

- Notify the station when a carrier is dead
- Make carriers' max charge as 10,000
- Recharging drone charges 1 dead carrier and flies back

Problems

- Piazza #606
- Takes a long time to charge a carrier
- Many dead carriers keep waiting
- 1 carrier may deliver every carrier's packages
    - Lifespan can decrease (overwork)

IMPLEMENTATION

Solutions

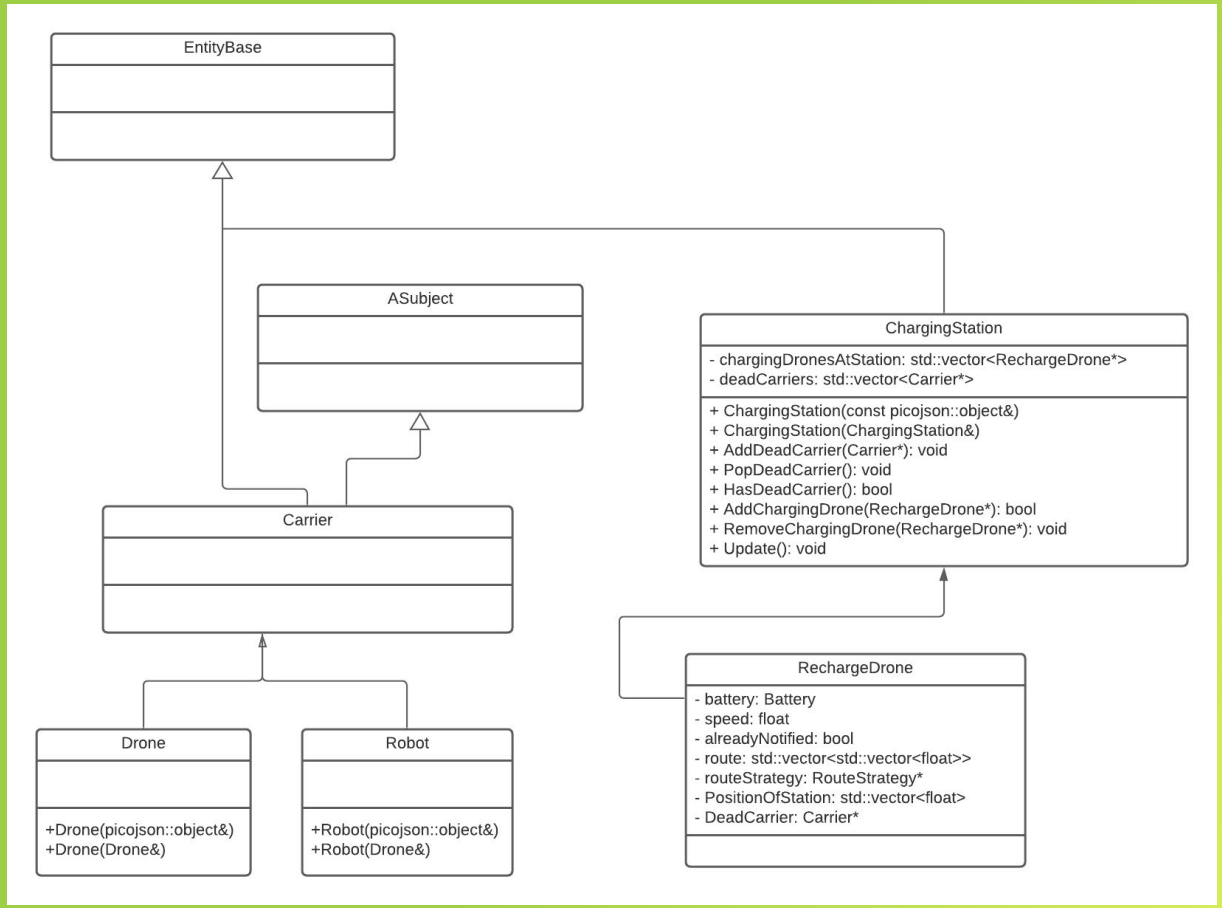
Carriers

- Initializes maximum battery capacity from JSON object

Recharging Drone

- Inherits EntityBase & ASubject
- Max charge: 20,000
- Charges 250x as fast
- Flies to the dead carrier “right away” by the Beeline path
- Back to station once battery capacity < 20%

NEW UML


EntityBase
ASubject
ChargingStation
- chargingDronesAtStation: std::vector<RechargeDrone*>
- deadCarriers: std::vector<Carrier*>
+ ChargingStation(const picojson::object&)
+ ChargingStation(ChargingStation&)
+ AddDeadCarrier(Carrier*): void
+ PopDeadCarrier(): void
+ HasDeadCarrier(): bool
+ AddChargingDrone(RechargeDrone*): bool
+ RemoveChargingDrone(RechargeDrone*): void
+ Update(): void
Carrier
RechargeDrone
- battery: Battery
- speed: float
- alreadyNotified: bool
- route: std::vector<std::vector<float>>
- routeStrategy: RouteStrategy*
- PositionOfStation: std::vector<float>
- DeadCarrier: Carrier*
Drone
+Drone(picojson::object&)
+Drone(Drone&)
Robot
+Robot(picojson::object&)
+Robot(Drone&)

IMPLEMENTATION

Solutions

Recharging Station

- **Inherits EntityBase**
- **Queue for dead carriers**

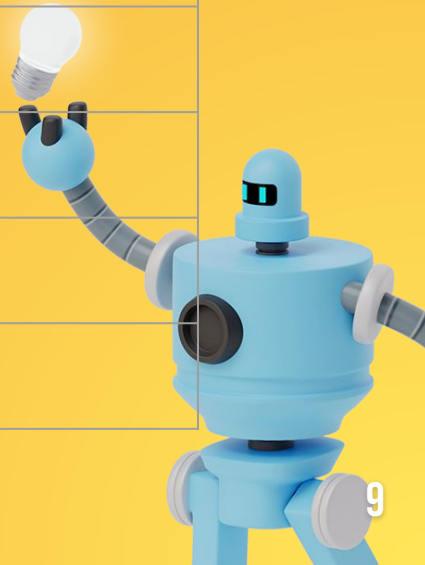
Keep track of dead carriers

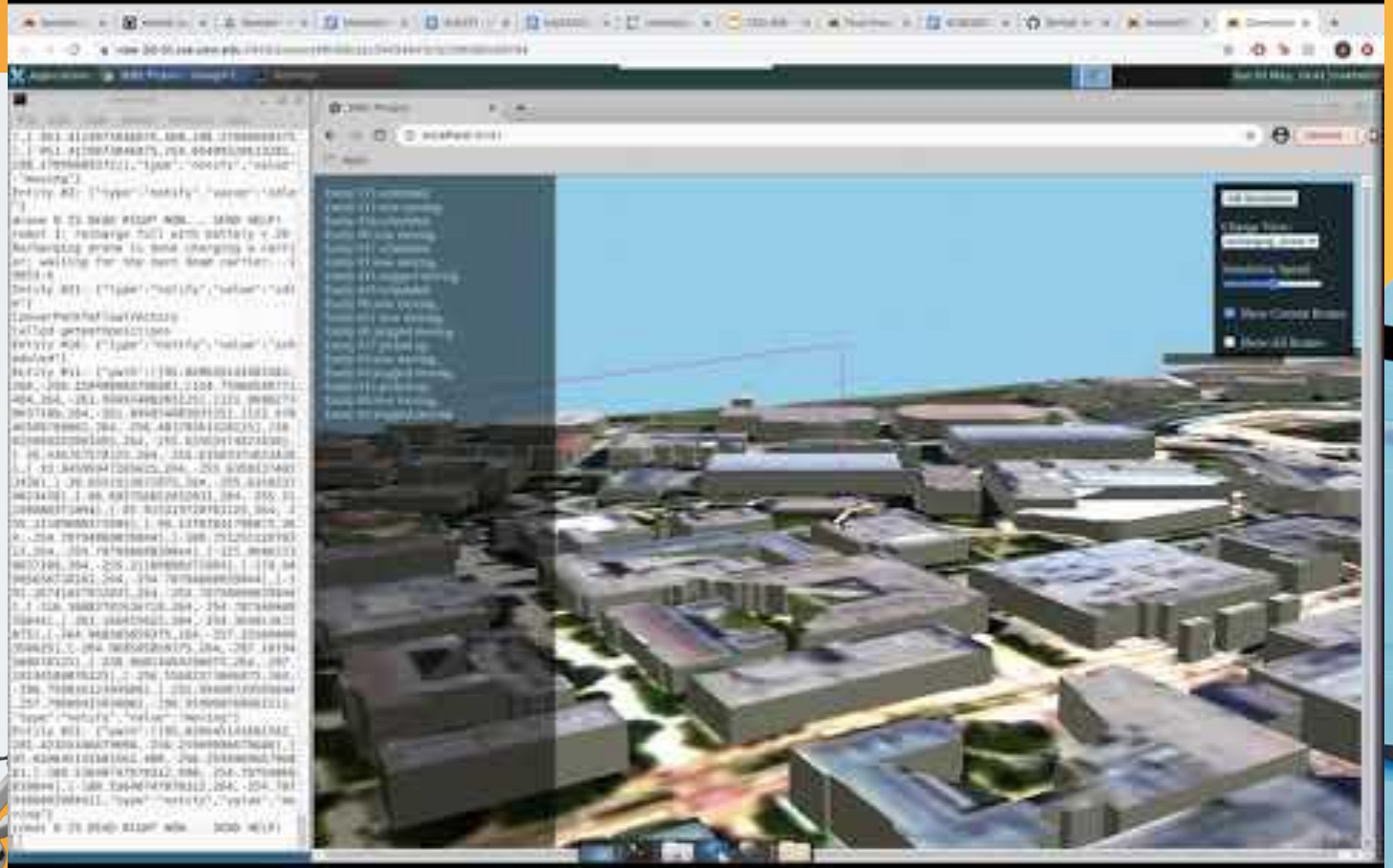
- **Use the carrier status to check if the battery is dead, if so, where is it (sky vs ground)**

Recharging

SCENE DESCRIPTION

ENTITY	MAX BATTERY CAPACITY	TYPE OF THE PATH
DRONE #1	60	SMART
DRONE #2	60	BEELINE
ROBOT #1	30	SMART
ROBOT #2	20	SMART
RECHARGING DRONE	20,000	BEELINE

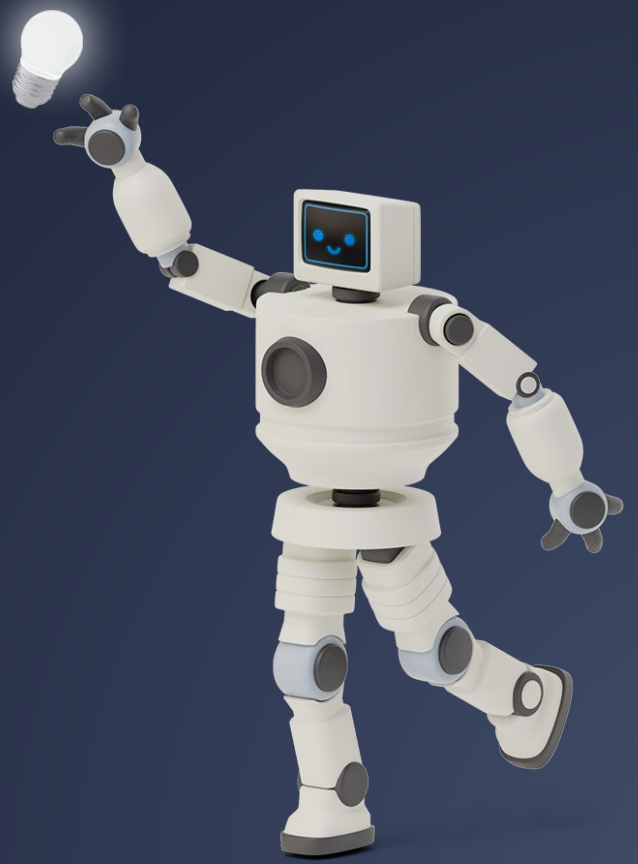




CONCLUSION

What did we learn?

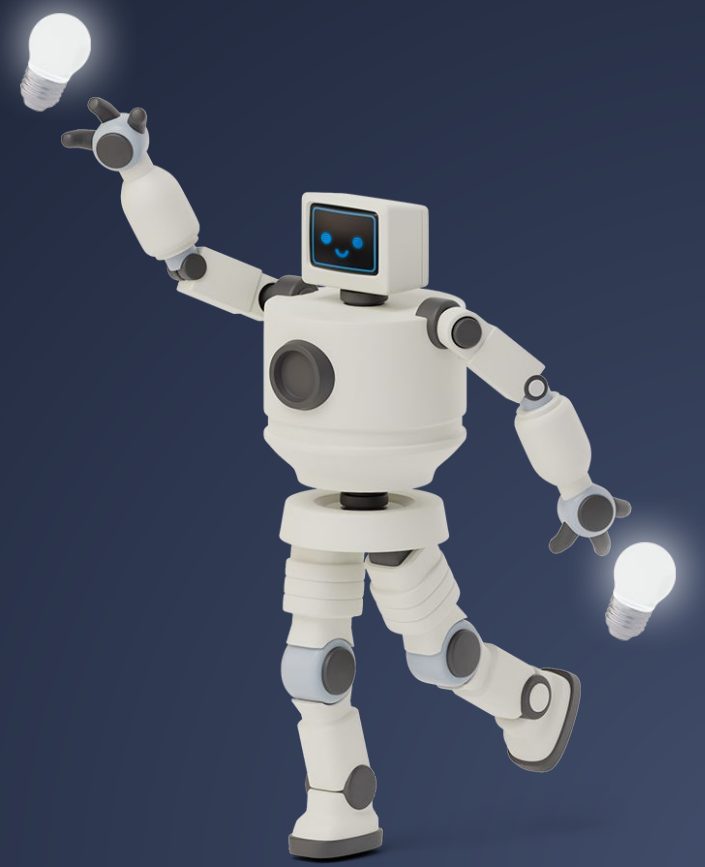
- Solidified how useful patterns are
- Learned how to work as a team when guiding our own goals
- Learned how effectively use GitHub as a team



CONCLUSION

What would we do differently?

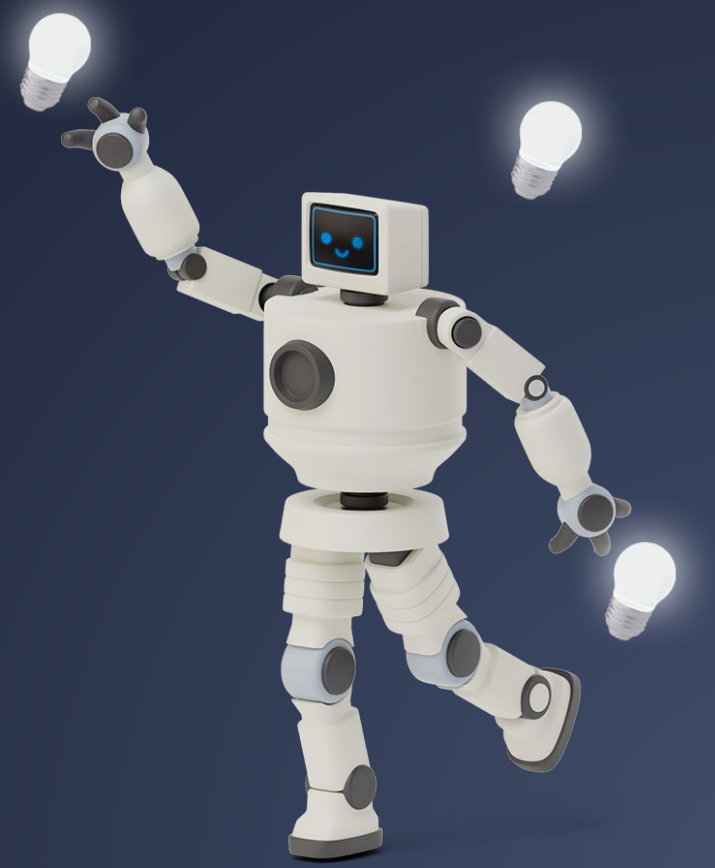
- Change the capacity at which recharge drone starts heading to station
- Utilise strategy pattern to make better choices for routes
- Multiple charging stations/ Multiple charging drones
  - Fly back to any closest recharging station
  - Charge dead carriers close to charging drone



CONCLUSION

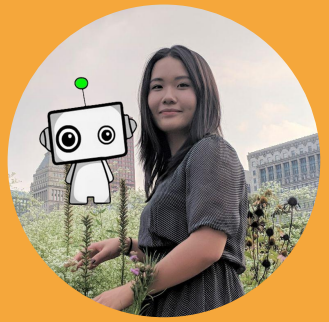
Does our approach work?

- Yes!!



ANY QUESTIONS?

TEAM PRESENTATION



Lin Huynh
Development Lead

Wiley Bui
Project Manager

Aunya Mukherjee
Scheduler

Dustin Zhang
Reporter