

Charger Implementation

Dante Schroeder

What is it?

- Charges drone battery
- Basic IEntity
- iPhone charger



Why is it important?

- Works with battery function
- Allows drones to function



Design pattern?

- Factory
- Easily create many chargers
- Place them in convenient locations
- Easy to implement



Source Code

```
1  #include "Charger.h"
2
3  Charger::Charger(JsonObject& obj) : details(obj) {
4      JSONArray pos(obj["position"]);
5      position = {pos[0], pos[1], pos[2]};
6  }
```

```
1  #include "ChargerFactory.h"
2
3  IEntity* ChargerFactory::CreateEntity(JsonObject& entity) {
4      std::string type = entity["type"];
5      if (type.compare("charger") == 0) {
6          std::cout << "Charger Created" << std::endl;
7          return new Charger(entity);
8      }
9      return nullptr;
10 }
```