Charger Implementation

Dante Schroeder

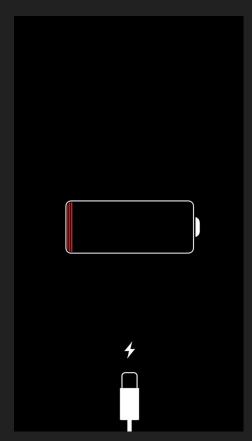
What is it?

- 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

```
#include "Charger.h"

Charger::Charger(JsonObject& obj) : details(obj) {
    JsonArray pos(obj["position"]);
    position = {pos[0], pos[1], pos[2]};
}
```

```
#include "ChargerFactory.h"

IEntity* ChargerFactory::CreateEntity(JsonObject& entity) {

std::string type = entity["type"];

if (type.compare("charger") == 0) {

std::cout << "Charger Created" << std::endl;

return new Charger(entity);

}

return nullptr;

}</pre>
```